DiscreteQuest — Flashcards

A Spark & Anvil printable flashcard deck. Quiz yourself on the DiscreteQuest cast's curriculum — one card at a time.

How to use these cards

Fold each sheet down the middle line so the answers are hidden. Read the **question** on the left, say your answer out loud, then **unfold** to check the right side. Testing yourself — then checking — is one of the most powerful ways to remember. Get one wrong? That's the best kind of practice: try it again tomorrow.

Want real flashcards? Cut along the fold line and the rows into cards — question on one side, answer on the other.

DiscreteQuest — Flashcards

- How to use these cards
- 1. Sets and Sorting
- 2. Counting and Patterns
- 3. Logic and Truth Tables
- 4. Graph Theory Basics
- 5. Combinatorics and Probability
- 6. Recursion and Sequences
- 7. Number Theory and Cryptography
- 8. Algorithm Design and Complexity
- 9. Elements & Principles Deep Dive
- 10. Cultural & Historical Context
- 11. Creative Process & Critique
- 12. Professional Practice & Careers
- 13. Introduction to Sets
- 14. Introduction to Sets
- 15. Introduction to Sets
- 16. Introduction to Sets
- Keep going
  - Keep exploring online

1. Sets and Sorting

✂ Question (fold →)	Answer
A PE teacher has students line up by birthday (January first, December last). She can either use bubble sort (compare neighbors) or selection sort (find the earliest birthday and move them to the front). With 25 students, which algorithm would likely finish faster in real life, and why?	**Selection sort would likely be faster because students can scan the whole line visually to find the earliest birthday, rather than making many small swaps.** — While both algorithms have similar computational complexity ($O(n^2)$), selection sort is often faster in practice for humans because we can visually scan a group quickly. Selection sort also minimizes the number of actual moves — each person moves at most once. Bubble sort requires many small swaps, which is physically slow with people moving around.
What is a set in mathematics?	**A set is a well-defined collection of distinct objects or elements.** — In mathematics, a set is simply a collection of distinct objects considered as a whole. Sets can contain numbers, letters, colors, or anything else. The important rule is that each element appears only once in a set — there are no duplicates.
Look at this Venn diagram: Circle A has {1, 3, 5}, the overlap has {7, 9}, and Circle B has {2, 4, 6}. What can you conclude about the odd numbers in this diagram?	**All odd numbers (1, 3, 5, 7, 9) are in Set A, while Set B contains only even numbers.** — Every odd number in the diagram (1, 3, 5, 7, 9) is inside Circle A — some exclusively in A, and some shared with B. Circle B's exclusive elements (2, 4, 6) are all even. This pattern suggests Set A might be defined as odd numbers while Set B is defined as even numbers, with 7 and 9 fitting both criteria.
If Set A = {1, 3, 5, 7} and Set B = {2, 4, 6, 8}, what do we call these two sets that share no elements?	**Sets that share no elements are called disjoint sets.** — Disjoint sets are sets that have no elements in common — their intersection is empty. Set A contains only odd numbers and Set B contains only even numbers, so they cannot share any elements. This concept is important in probability and data classification.
What does it mean for Set A to be a subset of Set B?	**Set A is a subset of Set B when every element in A is also found in B.** — A subset is a set entirely contained within another set. If A = {1, 2} and B = {1, 2, 3, 4}, then A is a subset of B (written $A \subseteq B$) because both 1 and 2 are in B. Every set is a subset of itself, and the empty set is a subset of every set.
In a card game, you pick up cards one at a time and slide each into its correct position among the cards already in your hand. Which sorting algorithm does this resemble?	**This resembles insertion sort, where each new element is inserted into its correct position in the sorted portion.** — Insertion sort works exactly like sorting cards in your hand. You take one card at a time and slide it into the correct position among the cards you've already sorted. It's efficient for small or nearly-sorted datasets, and it's the most intuitive sorting algorithm because people naturally use it.
If Set A = {1, 2, 3, 4, 5} and Set B = {3, 4, 5, 6, 7}, what is the intersection of A and B?	**The intersection of A and B is {3, 4, 5}.** — The intersection of two sets (written $A \cap B$) contains only the elements that appear in BOTH sets. Numbers 3, 4, and 5 are in both Set A and Set B. The intersection is like the overlapping region of a Venn diagram.
A teacher asks students to line up by height. She tells them: 'Compare yourself to the person next to you, and swap if the taller person is in front.' They repeat	**The teacher is using bubble sort, and it could be slow because it may need up to 435 comparisons (30 × 29 / 2) in the worst case.** — This is bubble sort — comparing adjacent elements and swapping until sorted. For 30 students, the worst case needs 30 × 29 / 2 = 435 comparisons. That's a lot of swapping!

this until everyone is in order. What sorting algorithm is the teacher using, and why might it be slow for a class of 30?	Faster algorithms like merge sort would only need about 150 comparisons (30 × $\log_2 30$), which is why computer scientists study algorithm efficiency.
What is the symbol ∈ used for in set notation?	**The symbol ∈ means 'is an element of' or 'belongs to' a set.** — The symbol ∈ is read as 'is an element of' or 'belongs to.' When we write 3 ∈ {1, 2, 3, 4}, we mean that 3 is a member of that set. Conversely, ∉ means 'is not an element of,' so 5 ∉ {1, 2, 3, 4}.
Your friend says bubble sort is the best sorting algorithm because it's the easiest to understand. Do you agree? What factors besides simplicity should we consider when choosing a sorting algorithm?	**While bubble sort is easy to understand, we should also consider speed, the size of the data, and whether the data is already partially sorted.** — Choosing an algorithm involves trade-offs: speed (time complexity), memory usage (space complexity), simplicity, and the nature of the data. Bubble sort is $O(n^2)$ and slow for large datasets but fine for small ones. Merge sort is $O(n \log n)$ and much faster for large datasets but uses more memory. The 'best' algorithm depends on the situation.
What is the empty set, and how is it written?	**The empty set is a set with no elements, written as { } or ∅.** — The empty set (also called the null set) contains no elements at all. It can be written as { } or with the special symbol ∅. Note that {0} is NOT empty — it contains the number zero. The empty set is a subset of every other set, which makes it special in mathematics.
Your class has 20 students who like pizza and 15 students who like tacos. 8 students like both. Use set operations to find how many students like pizza OR tacos.	**There are 27 students who like pizza or tacos, calculated as 20 + 15 - 8 = 27.** — This uses the inclusion-exclusion principle. If we add 20 + 15 = 35, we've counted the 8 students who like both pizza AND tacos twice. So we subtract 8 to avoid double-counting: 20 + 15 - 8 = 27. This is one of the most useful formulas in discrete mathematics.
Why does the set {3, 1, 2} equal the set {1, 2, 3}?	**The two sets are equal because sets only care about which elements they contain, not the order.** — Sets are unordered collections, meaning {3, 1, 2} and {1, 2, 3} contain the same elements and are therefore equal. This is different from sequences or lists where order matters. Two sets are equal if and only if they contain exactly the same elements.
A library sorts books by author last name. New books arrive in random order. Which sorting strategy would be most practical: re-sorting all books from scratch each time, or inserting each new book into its correct position?	**Inserting each new book into its correct position (insertion approach) is more practical because most books are already sorted.** — When most data is already sorted, insertion is far more efficient than re-sorting everything. This is why insertion sort performs well on nearly-sorted data. Libraries, filing systems, and databases all use this principle — maintain sorted order incrementally rather than rebuilding from scratch.
In a Venn diagram with two overlapping circles, where would you place an item that belongs to both Set A and Set B?	**An item belonging to both sets goes in the overlapping region where the two circles intersect.** — In a Venn diagram, the overlapping region (called the intersection) shows elements that belong to both sets simultaneously. Items only in Set A go in the left-only area, items only in Set B go in the right-only area, and shared items go in the middle overlap.
If Set A = {1, 3, 5} and Set B = {2, 3, 4}, what is A ∩ B (the intersection)?	**{3}** — See the correct answer above for the explanation.
You have the set of colors {red,	**The subset of primary colors is {red, blue, yellow}.** — The primary colors in traditional color theory are red, blue, and yellow. Since all three of these are in

blue, green, yellow, purple}. Create a subset of primary colors from this set.	our original set, {red, blue, yellow} is a valid subset. Note that RGB (red, green, blue) are the primary colors of light, not paint — context matters in classification!
Invent a new sorting game for your classmates. Describe what objects you'd sort, what property you'd sort by, and which sorting algorithm (bubble, selection, or insertion) your game would use. Explain why you chose that algorithm.	**A good game might sort playing cards by number using insertion sort, since players naturally pick up cards one at a time and place them in order, making it intuitive and fun.** — Designing a sorting game requires creativity and understanding of algorithms. Insertion sort works well for card games because it mirrors how people naturally organize. Selection sort works for relay races where teams send someone to find the 'minimum' item. Bubble sort works for partner activities where neighbors compare. The best game choice depends on what's fun AND educational.
Design a set-based system to organize your school's after-school clubs. Define at least three sets and show how union and intersection could help answer questions about student participation.	**You could define sets like Sports = {students in sports}, Arts = {students in art clubs}, and Academic = {students in academic clubs}, then use intersection to find students in multiple clubs and union to find all participating students.** — Sets are perfect for modeling group membership. Sports ∩ Arts finds multi-activity students who might have scheduling conflicts. Sports ∪ Arts ∪ Academic finds all students in at least one club. The complement (students NOT in any club) identifies those who might want recruiting. This is how databases, social networks, and analytics systems all use set theory.
If Set A = {apple, banana, cherry} and Set B = {banana, date, elderberry}, what is the union of A and B?	**The union of A and B is {apple, banana, cherry, date, elderberry}.** — The union of two sets (written A ∪ B) combines all elements from both sets, listing each element only once. Even though banana appears in both sets, it's listed just once in the union. Think of union as 'everything from either set.'
You have 5 books to arrange on a shelf. If you sort them by title alphabetically, how many comparisons does it take using bubble sort in the worst case?	**Bubble sort needs at most 10 comparisons to sort 5 items, calculated as 4 + 3 + 2 + 1 = 10.** — Bubble sort compares adjacent elements and swaps them if needed. With 5 items, the first pass needs 4 comparisons, the second needs 3, the third needs 2, and the last needs 1. Total: 4 + 3 + 2 + 1 = 10. This pattern gives us n(n-1)/2 comparisons for n items.
Sort the list [5, 2, 8, 1, 9] using selection sort. What does the list look like after the first pass?	**After the first pass, the list is [1, 2, 8, 5, 9] because 1 (the smallest) swaps with 5 (the first element).** — Selection sort works by finding the smallest element in the unsorted portion and swapping it into the next sorted position. The smallest element in [5, 2, 8, 1, 9] is 1, which is at position 4. We swap it with position 1 (value 5), giving us [1, 2, 8, 5, 9]. Now position 1 is sorted.
In a class survey, Set M = students who like math, Set S = students who like science, and Set A = students who like art. If 5 students like all three subjects, where would they appear in a three-circle Venn diagram?	**Students who like all three subjects appear in the center region where all three circles overlap.** — A three-circle Venn diagram has 7 distinct regions: 3 exclusive regions, 3 two-way overlaps, and 1 three-way overlap in the center. Students who like all three subjects belong to the intersection M ∩ S ∩ A, which is the center region. Three-circle Venn diagrams are powerful tools for analyzing complex relationships.
A student claims that if Set A has 4 elements and Set B has 6 elements, then A ∪ B must have exactly 10 elements. Is this always true? Explain why or why not.	**This is not always true because if A and B share elements, the union will have fewer than 10 elements.** — The union has exactly 10 elements only if A and B are disjoint (share nothing). If they share elements, the union is smaller. For example, if A = {1, 2, 3, 4} and B = {3, 4, 5, 6, 7, 8}, the union is {1, 2, 3, 4, 5, 6, 7, 8} with 8 elements, not 10. The formula |A ∪ B| = |A| + |B| - |A ∩ B| accounts for this.

How do we write the set of all even numbers between 1 and 11 using set notation with curly braces?	**The set of even numbers between 1 and 11 is written as {2, 4, 6, 8, 10}.** — Sets are written using curly braces { } with elements separated by commas. The even numbers between 1 and 11 are 2, 4, 6, 8, and 10. Notice we don't include 1 or 11 (they're odd), and we don't go past 10 since 12 is beyond 11.

2. Counting and Patterns

✂ Question (fold →)	Answer
Why do passwords get exponentially harder to guess as they get longer? Compare a 4-character password to an 8-character password using only lowercase letters (26 options).	**A 4-character password has $26^4 \approx$ 457,000 possibilities, while an 8-character password has $26^8 \approx$ 209 billion, making it about 457,000 times harder to guess.** — 26^4 = 456,976 and 26^8 = 208,827,064,576. Doubling the password length doesn't double the difficulty — it SQUARES it ($26^8 = (26^4)^2$). This exponential growth is why cybersecurity experts recommend long passwords. Each additional character multiplies the total possibilities by 26, making brute-force guessing impractical.
Create a counting problem about planning a class trip. Include at least three independent choices that must be made, and calculate the total number of possible trip plans.	**A good problem might be: Choose a destination (4 options), a bus type (3 options), and a lunch plan (5 options), giving 4 × 3 × 5 = 60 different trip plans.** — The multiplication principle extends to any number of independent choices. A trip plan might involve: destination (museum, park, zoo, factory = 4), transportation (school bus, coach bus, van = 3), and lunch (pizza, sandwiches, wraps, salad bar, hot dogs = 5). Total: 4 × 3 × 5 = 60 unique trip plans. Creating your own counting problems deepens understanding of the underlying structure.
A student says there are 24 ways to arrange the letters in the word MATH. Another student says there are also 24 ways to arrange MEET. Who is correct, and why?	**The first student is correct (MATH has 4! = 24 arrangements), but MEET has only 12 because the two E's are identical, giving 4!/2! = 12.** — MATH has 4 distinct letters, so there are 4! = 24 unique arrangements. MEET has a repeated E, so swapping the two E's doesn't create a new arrangement. We divide by 2! to account for the identical letters: 4!/2! = 24/2 = 12. This principle extends: MISSISSIPPI has 11!/(4!4!2!) = 34,650 arrangements.
You need a 4-digit PIN using digits 0-9. How many PINs are possible if repetition is allowed?	**There are 10,000 possible PINs, calculated as 10 × 10 × 10 × 10 = 10,000.** — With repetition allowed, each of the 4 positions has 10 independent choices (0-9), giving 10^4 = 10,000 possible PINs (from 0000 to 9999). PINs starting with 0 are valid! Without repetition, it would be 10 × 9 × 8 × 7 = 5,040 PINs — nearly half as many.
Design a number sequence that follows a rule you create. Write the first 6 terms and describe the rule so a classmate could continue the pattern.	**A good sequence might be: 2, 6, 12, 20, 30, 42 where the rule is 'each term is n × (n + 1)' or equivalently 'the differences increase by 2 each time: 4, 6, 8, 10, 12.'** — Creating sequences helps you understand patterns deeply. The sequence 2, 6, 12, 20, 30, 42 uses the rule n(n+1): 1×2=2, 2×3=6, 3×4=12, 4×5=20, 5×6=30, 6×7=42. The next term would be 7×8=56. Notice these are twice the triangular numbers! Many interesting sequences hide connections to other mathematical patterns.
In how many ways can 6 runners finish a race in first, second, and third place?	**There are 120 ways, calculated as 6 × 5 × 4 = 120.** — Any of the 6 runners could finish first. Once first is determined, 5 remain for second place. Then 4 remain for third. By the multiplication principle: 6 × 5 × 4 = 120 ways. Note that 6! = 720 would be the answer if we needed ALL 6 finishing positions, not just the top 3.
Find the pattern and fill in the blank: 1, 4, 9, 16, 25, __, 49.	**The missing number is 36, because these are perfect squares: 1^2, 2^2, 3^2, 4^2, 5^2, 6^2, 7^2.** — These are perfect squares: $1^2 = 1$, $2^2 = 4$, $3^2 = 9$, $4^2 = 16$, $5^2 = 25$, $6^2 = 36$, $7^2 = 49$. The missing value is $6^2 = 36$. Notice the differences between consecutive squares form their own pattern: 3, 5, 7, 9, 11, 13 — always increasing by 2. This is because $(n+1)^2 - n^2 = 2n + 1$.

A school password must be exactly 3 characters: the first must be a letter (A-Z), the second must be a digit (0-9), and the third must be a special symbol chosen from {!, @, #, $, %}. A security expert says this is less secure than a 3-character password using any of these 41 characters in any position. Is the expert correct?	**Yes, the expert is correct. The restricted format gives 26 × 10 × 5 = 1,300 passwords, while the unrestricted format gives 41 × 41 × 41 = 68,921 passwords.** — The restricted format has only 26 × 10 × 5 = 1,300 options because each position is limited. The unrestricted format has 41^3 = 68,921 options — about 53 times more secure. While the restricted format seems 'stronger' because it forces different character types, it actually tells attackers exactly what to expect in each position, dramatically reducing the search space.
What does 5! (five factorial) equal?	**Five factorial equals 120, calculated as 5 × 4 × 3 × 2 × 1 = 120.** — The factorial of a number n (written n!) is the product of all positive integers from 1 to n. So 5! = 5 × 4 × 3 × 2 × 1 = 120. Factorials grow incredibly fast: 10! = 3,628,800 and 20! is over 2 quintillion. Factorials are used to count permutations — there are n! ways to arrange n distinct objects.
What is the difference between a permutation and a combination?	**In a permutation the order of selection matters, while in a combination the order does not matter.** — Permutations care about order (lock codes, race placements, ranked positions), while combinations don't (choosing team members, picking pizza toppings). Choosing president then vice-president from 5 people gives more permutations than just choosing 2 people for a committee, because order distinguishes the roles.
Your 4-digit bike lock allows digits 0-9 with repetition. Your friend's lock also has 4 positions but uses letters A-Z (26 options) without repetition. Whose lock has more possible combinations?	**Your friend's letter lock has more combinations: 26 × 25 × 24 × 23 = 358,800, compared to your 10 × 10 × 10 × 10 = 10,000.** — Your number lock: 10^4 = 10,000 combinations. Your friend's letter lock: 26 × 25 × 24 × 23 = 358,800 combinations. Despite not allowing repetition, the letter lock wins by a factor of 35 because the larger alphabet (26 vs 10) more than compensates. This shows that the size of the symbol set matters more than whether repetition is allowed.
What comes next in the pattern: 2, 6, 18, 54, ...?	**162** — See the correct answer above for the explanation.
How many different ways can you arrange the letters A, B, and C in a row?	**6** — See the correct answer above for the explanation.
Why is 4! (24) the number of ways to arrange 4 different books on a shelf?	**For the first position you have 4 choices, then 3 for the second, 2 for the third, and 1 for the last, giving 4 × 3 × 2 × 1 = 24.** — This is the multiplication principle in action. With 4 books: 4 choices for position 1, then 3 remaining for position 2, then 2 for position 3, and 1 for the last. The product 4 × 3 × 2 × 1 = 24 counts every possible arrangement. This is why n! counts the permutations of n distinct objects.
If a password must be 2 digits long and each digit is 0-9, how many possible passwords are there?	**100** — See the correct answer above for the explanation.
If you have 3 shirts and 4 pairs of pants, how many different outfits can you make?	**You can make 12 different outfits, calculated as 3 × 4 = 12.** — The multiplication principle (also called the counting principle) states that if you have m choices for one decision and n choices for another independent decision, the total number of combined outcomes is m × n. With 3 shirts and 4 pants: 3 × 4 = 12 outfits.

Look at this pattern of triangular numbers: 1, 3, 6, 10, 15, 21, 28. What is the rule for finding the next triangular number, and why are they called 'triangular'?	**Each triangular number adds one more than the previous gap (add 2, add 3, add 4, etc.), and they're called triangular because they represent dots arranged in equilateral triangles.** — Triangular numbers follow the pattern: $T(n) = 1 + 2 + 3 + ... + n$. The differences between consecutive terms increase by 1 each time (2, 3, 4, 5, 6, 7). They're called triangular because you can arrange that many dots into a perfect equilateral triangle: row 1 has 1 dot, row 2 has 2, row 3 has 3, and so on. The formula is $T(n) = n(n+1)/2$.
What does the word 'permutation' mean in mathematics?	**A permutation is an arrangement of objects where the order matters.** — A permutation is an ordered arrangement. The sequence A-B-C is a different permutation than C-B-A, even though they use the same letters. This contrasts with combinations, where order doesn't matter. Permutations are used in passwords, lock codes, race finishes, and seating arrangements.
What is 0! (zero factorial)?	**Zero factorial equals 1 by mathematical convention.** — By convention, $0! = 1$. While this seems counterintuitive, it makes mathematical sense: there is exactly one way to arrange zero objects (the empty arrangement). This convention also ensures that formulas like the binomial coefficient $n!/(k!(n-k)!)$ work correctly when $k = 0$ or $k = n$.
Your teacher says the Fibonacci sequence (1, 1, 2, 3, 5, 8, 13, ...) appears everywhere in nature. A classmate says this is just a coincidence. Who do you think makes a stronger argument, and what evidence supports your view?	**The teacher makes a stronger argument because Fibonacci numbers appear in sunflower seed spirals, pinecone scales, flower petals, and shell curves, suggesting a deep connection to natural growth patterns.** — The teacher has strong evidence. Fibonacci numbers appear in: sunflower seed spirals (usually 34 and 55), pine cone scales (8 and 13), flower petals (lilies have 3, buttercups 5, daisies often 34 or 55), and nautilus shell growth. This isn't coincidence — these numbers emerge from efficient growth and packing algorithms in biological systems.
A pizza shop offers 3 sizes, 2 crust types, and 5 toppings. If you pick one of each, how many different single-topping pizzas are possible?	**There are 30 different single-topping pizzas, calculated as $3 \times 2 \times 5 = 30$.** — Each pizza requires three independent choices: size (3 options), crust (2 options), and topping (5 options). The multiplication principle gives us $3 \times 2 \times 5 = 30$ different combinations. This extends to any number of independent choices — just keep multiplying!
Why does a 3-digit lock with digits 0-9 have 1,000 possible codes instead of 30?	**Each of the 3 positions can independently be any of 10 digits, so there are $10 \times 10 \times 10 = 1{,}000$ codes.** — Since digits can repeat, each of the 3 positions has 10 independent choices (0-9). By the multiplication principle: $10 \times 10 \times 10 = 1{,}000$ possible codes (000 through 999). If repetition were NOT allowed, there would be only $10 \times 9 \times 8 = 720$ codes. Whether repetition is allowed dramatically changes the count.
What pattern do you see in this sequence: 1, 1, 2, 3, 5, 8, 13, ...? What are the next two numbers?	**Each number is the sum of the two before it (the Fibonacci sequence), so the next two numbers are 21 and 34.** — This is the Fibonacci sequence, one of the most famous patterns in mathematics. Each term is the sum of the two preceding terms: $8 + 13 = 21$, and $13 + 21 = 34$. Fibonacci numbers appear in nature — sunflower spirals, pine cone patterns, and shell curves all follow this sequence.
How many 3-letter 'words' (real or nonsense) can you make from the letters A, B, C, D, E if no letter can be	**You can make 60 three-letter words, calculated as $5 \times 4 \times 3 = 60$.** — Without repetition: 5 choices for the first letter, 4 for the second (one is used), and 3 for the third. So $5 \times 4 \times 3 = 60$ arrangements. This is written as $P(5,3)$ or $5!/(5-3)! = 120/2 = 60$. If repetition were allowed, it would be 5

repeated?	× 5 × 5 = 125.
A school lunch offers 4 main dishes, 3 side dishes, and 2 drinks. The school wants to know how many different meal combinations exist. If they add one more drink option, how does the total change?	**Originally there are 4 × 3 × 2 = 24 meals. Adding one drink makes it 4 × 3 × 3 = 36 meals — an increase of 12.** — Before: 4 × 3 × 2 = 24 meals. After: 4 × 3 × 3 = 36 meals. The increase is 12, which equals 4 × 3 × 1 — the number of main+side combinations times the one new drink. Adding a single option to one category multiplies through all the other categories, which is why menus with many categories grow so quickly.

3. Logic and Truth Tables

✂ Question (fold →)	Answer
Compare the truth tables for P OR Q and P AND Q. In how many of the 4 input combinations do they give different results?	**They give different results in 2 of the 4 combinations: when one input is true and the other is false.** — AND and OR agree on (T,T)→both true, and (F,F)→both false. They differ on (T,F): OR=true, AND=false, and (F,T): OR=true, AND=false. So they differ in exactly 2 of 4 cases. This shows AND is stricter than OR — AND requires unanimity while OR only needs one vote.
In a video game, a treasure chest opens when the player 'hasKey AND (level >= 5 OR isPremium).' If a player has a key and is level 3 but is a premium member, does the chest open?	**Yes, the chest opens because hasKey is true, and isPremium is true which makes the OR clause true, so the whole AND expression is true.** — Work from the inside out. First: (level >= 5 OR isPremium) = (false OR true) = true. Then: hasKey AND true = true AND true = true. The chest opens! The OR inside the parentheses means either condition suffices. This is exactly how game programmers and app developers write conditional logic.
Why is the statement 'If it's Tuesday, then I have math class' different from 'If I have math class, then it's Tuesday'?	**They're different because the second is the converse of the first, and a statement and its converse can have different truth values.** — The original says Tuesday guarantees math class. The converse says math class guarantees it's Tuesday — but you might have math on other days too! A statement and its converse are NOT logically equivalent. This is a common reasoning error. The contrapositive ('If I don't have math, then it's not Tuesday') IS equivalent to the original.
In logic, what does the AND operation mean?	**The AND operation is true only when both statements connected by it are true.** — The AND operation (also written as ∧) requires both conditions to be true. 'It's sunny AND it's warm' is only true when BOTH conditions hold. If either one is false, the whole AND statement is false. This is used everywhere in programming: 'if (age >= 13 AND hasPermission)' requires both conditions.
What does the NOT operation do to a logical statement?	**The NOT operation reverses the truth value — it turns true into false and false into true.** — NOT (also written as ¬ or !) is the simplest logical operation — it just flips the truth value. If P is true, then NOT P is false, and vice versa. In programming, this is the '!' operator: if (isLoggedIn) checks for true, while if (!isLoggedIn) checks for false. NOT is essential for expressing negation in logic circuits, code, and mathematical proofs.
How does exclusive OR (XOR) differ from regular OR?	**XOR is true when exactly one input is true, but false when both are true, while regular OR is true when both are true.** — Regular OR (inclusive) is true when one or both inputs are true. XOR (exclusive OR) is true when exactly one input is true, but false when both are true. The menu example 'soup or salad' is XOR — you pick one. The security rule 'enter with a badge or sign in' is regular OR — you can do both. XOR is crucial in encryption and error detection.
A weather app says: 'If temperature > 90°F, then display heat warning.' The contrapositive is: 'If no heat warning is displayed, then temperature ≤ 90°F.' Are both statements always true at the same time?	**Yes, a conditional statement and its contrapositive are logically equivalent — they always have the same truth value.** — A statement and its contrapositive are ALWAYS logically equivalent. If 'A → B' is true, then 'NOT B → NOT A' must also be true. This is different from the converse ('if heat warning, then temp > 90') which might be false — maybe the system has a bug. The contrapositive is a powerful reasoning tool in mathematics, programming, and everyday logic.

Your friend argues: 'All dogs are animals. My cat is an animal. Therefore, my cat is a dog.' What logical error is your friend making?	**Your friend is committing the fallacy of affirming the consequent — just because all dogs are animals doesn't mean all animals are dogs.** — This is the fallacy of affirming the consequent. 'If dog, then animal' does NOT mean 'if animal, then dog.' Dogs are a subset of animals, but cats, birds, and fish are also animals. Confusing a statement with its converse is one of the most common logical errors in everyday reasoning, advertising, and even scientific claims.
What is the negation of the statement 'All birds can fly'?	**Some birds cannot fly** — See the correct answer above for the explanation.
What is a truth table?	**A truth table is a chart that shows every possible combination of true/false inputs and the resulting output for a logical expression.** — A truth table systematically lists every possible combination of input values and shows what the output is for each. With 2 inputs, there are 4 rows (TT, TF, FT, FF). With 3 inputs, there are 8 rows. Truth tables are used to verify logical arguments, design digital circuits, and debug computer programs.
Your friend claims: 'If NOT (P AND Q) is true, then both P and Q must be false.' Build a truth table to evaluate this claim.	**The claim is false. NOT (P AND Q) is true whenever at least one is false — it includes (T,F), (F,T), and (F,F), not just (F,F).** — Truth table: P AND Q is true only for (T,T). NOT (P AND Q) flips this: false for (T,T), true for (T,F), (F,T), and (F,F). So the claim is wrong — NOT (P AND Q) is true in THREE cases, not just when both are false. By De Morgan's Law, NOT (P AND Q) = (NOT P) OR (NOT Q), which is a much weaker condition than requiring both to be false.
What is a conditional statement in logic?	**A conditional statement is an 'if-then' statement, written as 'if P, then Q' where P is the hypothesis and Q is the conclusion.** — Conditional statements (if P then Q, written $P \rightarrow Q$) are fundamental to logic and programming. P is called the hypothesis (or antecedent) and Q is the conclusion (or consequent). Surprisingly, 'if P then Q' is only false when P is true and Q is false. If P is false, the conditional is automatically true regardless of Q.
De Morgan's Law says NOT (P AND Q) = (NOT P) OR (NOT Q). Can you explain why this makes intuitive sense using a real-world example?	**If it's NOT the case that 'you passed the test AND did the homework,' that means you either didn't pass the test OR didn't do the homework (or both).** — De Morgan's Law captures everyday reasoning perfectly. If someone says 'You didn't BOTH clean your room and do your homework,' it means at least one wasn't done — you didn't clean OR you didn't do homework. Similarly, NOT (P OR Q) = (NOT P) AND (NOT Q): 'You didn't do either' means you didn't do the first AND you didn't do the second.
In logic, what does the OR operation mean?	**The OR operation is true when at least one of the statements connected by it is true.** — The OR operation (also written as $\vee$) is true when at least one condition is true — including when both are true. 'I'll bring an umbrella if it's raining OR cloudy' means any rain, any clouds, or both will trigger the action. Note: logical OR differs from everyday English 'or,' which often means 'one or the other but not both' (called exclusive OR).
Create a logic puzzle for your classmates. Write three if-then clues that, when combined, allow the solver to determine which student (Amy, Ben, or Cal) plays which instrument (drums, flute, or guitar).	**A good puzzle might be: (1) If Amy plays drums, then Ben plays guitar. (2) Cal does NOT play flute. (3) If Ben does NOT play drums, then Amy plays flute. From clue 2, Cal plays drums or guitar. Testing possibilities leads to a unique solution.** — Logic puzzles use conditional reasoning and elimination. From clue 2: Cal plays drums or guitar. Try Cal = drums: then from clue 1's contrapositive, we explore possibilities. These puzzles develop systematic reasoning skills. The key is that each clue eliminates possibilities until only one consistent assignment remains — this is the essence of logical deduction.

Complete the truth table for P AND Q. When P = True and Q = False, what is the result?	**When P is true and Q is false, P AND Q is false because AND requires both inputs to be true.** — AND produces true ONLY when both inputs are true. The complete truth table for P AND Q is: (T,T)→T, (T,F)→F, (F,T)→F, (F,F)→F. Only one of the four combinations gives true. This is why AND is considered a 'strict' operation — it's very hard to make an AND statement true.
A school rule says: 'Students can use the computer lab if they have finished homework AND it is not lunchtime.' Express this as a logical expression and evaluate it when homework is done and it IS lunchtime.	**The expression is 'finishedHomework AND NOT lunchtime.' When homework is done (true) and it is lunchtime (true), NOT lunchtime is false, so the result is false — the student cannot use the lab.** — Step by step: finishedHomework = true, lunchtime = true, NOT lunchtime = false. Then: true AND false = false. The student cannot use the lab even though homework is done, because it's lunchtime. Both conditions of the AND must be satisfied. This is how access control systems work in buildings, websites, and apps.
An advertisement claims: 'Professional athletes use our shoes. If you use our shoes, you'll perform like a professional.' Is this a valid logical argument? Explain the flaw.	**This is not valid — it's the fallacy of affirming the consequent. Professional athletes using the shoes doesn't mean the shoes cause professional performance.** — The ad commits two logical errors: affirming the consequent (professionals use X, therefore X causes professionalism) and confusing correlation with causation. Professional athletes succeed due to years of training, genetics, coaching — not shoes. Recognizing these fallacies in advertising is a critical thinking skill that logic training develops.
If P is true and Q is false, what is the truth value of P AND Q?	**False** — See the correct answer above for the explanation.
In a truth table, how many rows are needed to evaluate a compound statement with 3 variables?	**8** — See the correct answer above for the explanation.
Use a truth table to verify whether 'P AND (P OR Q)' always equals P.	**Yes, P AND (P OR Q) always equals P. When P is true, the expression is true; when P is false, the expression is false — matching P exactly.** — Row by row: (T,T): T AND (T OR T) = T AND T = T = P✓. (T,F): T AND (T OR F) = T AND T = T = P✓. (F,T): F AND (F OR T) = F AND T = F = P✓. (F,F): F AND (F OR F) = F AND F = F = P✓. The result matches P in every row, proving the absorption law: P AND (P OR Q) ≡ P.
Design a set of rules for a simple robot using AND, OR, and NOT. The robot should: move forward if the path is clear AND battery is charged, turn left if there's an obstacle AND the left path is clear, and stop if the battery is NOT charged.	**The rules could be: moveForward = clearAhead AND charged; turnLeft = obstacleAhead AND clearLeft; stop = NOT charged. These three rules cover the basic scenarios using logical operations.** — These rules demonstrate how logical operations control real systems. The AND operations ensure multiple conditions are checked simultaneously. The NOT operation creates a safety override. In real robotics, these Boolean expressions form the basis of decision-making. Priority matters too — the stop rule (NOT charged) should override the others for safety.
A programmer writes: 'if (age >= 13 OR hasParentPermission) AND accountActive.' Is this the right logic for 'users 13 and older can access the site, or younger users with parent	**Yes, this logic correctly captures the requirement. Users need an active account AND must either be 13+ or have parental permission.** — Let's verify: A 10-year-old with permission and active account: (false OR true) AND true = true ✓. A 15-year-old without permission, active: (true OR false) AND true = true ✓. A 14-year-old, active but deactivated account: (true OR false) AND false = false ✓. The parentheses correctly group the age/permission OR, then AND it with the account

permission, but only if their account is active'?	status requirement.
Build a truth table for NOT (P OR Q). How many rows does it have, and when is the result true?	**The truth table has 4 rows (for TT, TF, FT, FF), and NOT (P OR Q) is true only when both P and Q are false.** — With 2 variables, the truth table has $2^2 = 4$ rows. P OR Q is true for (T,T), (T,F), (F,T) and false only for (F,F). Applying NOT flips everything: NOT(P OR Q) is false for the first three and true only for (F,F). This equals (NOT P) AND (NOT Q) — that's De Morgan's Law in action!
What does it mean when two logical expressions are 'logically equivalent'?	**Two expressions are logically equivalent when they have identical truth tables — they produce the same output for every possible input combination.** — Logical equivalence means two expressions always agree. For example, 'NOT (P AND Q)' is logically equivalent to '(NOT P) OR (NOT Q)' — this is De Morgan's Law. You can verify by building truth tables for both and checking that every row matches. Logical equivalence is how mathematicians simplify complex expressions.

4. Graph Theory Basics

✂ Question (fold →)	Answer
In a weighted graph of cities, the edges show distances in miles: A-B = 10, B-C = 15, A-C = 30, B-D = 20, C-D = 5. What is the shortest path from A to D?	**The shortest path from A to D is A → B → C → D with a total distance of 30 miles (10 + 15 + 5).** — Possible paths: A→B→D = 10+20 = 30, A→C→D = 30+5 = 35, A→B→C→D = 10+15+5 = 30. Both A→B→D and A→B→C→D have length 30, but A→B→C→D routes through more cities. The key insight is that the direct-looking path isn't always shortest — this is why GPS algorithms check many routes. Dijkstra's algorithm solves this efficiently.
What is a cycle in a graph, and how is it different from a path?	**A cycle is a path that starts and ends at the same vertex, forming a closed loop.** — A cycle is a closed path — it starts at a vertex, visits other vertices without repeating, and returns to the start. A triangle (3 vertices connected in a loop) is the smallest cycle. Cycles are important in detecting deadlocks in computer systems, finding loops in circuits, and identifying redundant connections in networks.
Draw a graph representing the friendships in a group: Amy is friends with Ben and Cal. Ben is friends with Amy and Dana. Cal is friends with Amy only. Dana is friends with Ben only. How many vertices and edges does this graph have?	**The graph has 4 vertices (Amy, Ben, Cal, Dana) and 3 edges (Amy-Ben, Amy-Cal, Ben-Dana).** — Four people = 4 vertices. The friendships are: Amy-Ben, Amy-Cal, and Ben-Dana = 3 edges. Note that Amy is friends with Ben is the SAME edge as Ben is friends with Amy (undirected graph). Amy has degree 2 (connected to Ben and Cal), Ben has degree 2 (Amy and Dana), Cal has degree 1, and Dana has degree 1. The sum of degrees (2+2+1+1=6) equals twice the edges (2×3=6) — the Handshaking Lemma!
What is a cycle in a graph?	**A path that starts and ends at the same vertex without repeating edges** — See the correct answer above for the explanation.
The Handshaking Lemma says the sum of all vertex degrees in a graph equals twice the number of edges. Use this to determine if a graph can exist with 4 vertices having degrees 3, 2, 2, and 1.	**No, this graph cannot exist because the sum of degrees is 3 + 2 + 2 + 1 = 8, and 8/2 = 4 edges, but a vertex of degree 3 must connect to 3 others, and the degree-1 vertex can only connect to one — the constraints conflict.** — The degree sum is 8 (even), suggesting 4 edges could work. But let's try: vertex A (degree 3) connects to B, C, and D. Then B already has degree 1 (from A), C has degree 1, D has degree 1. We need B at degree 2, so B needs one more edge. B connects to C: B is now degree 2 ✓, C is degree 2 ✓. D is at degree 1 ✓, A is at degree 3 ✓. Actually, it DOES work! The graph A-B, A-C, A-D, B-C is valid.
In graph theory, what is the degree of a vertex?	**The number of edges connected to it** — See the correct answer above for the explanation.
In graph theory, what is a vertex (also called a node)?	**A vertex is a point in a graph that represents an object, person, or location.** — A vertex (plural: vertices) is a fundamental unit of a graph, representing an entity. In a friendship graph, each person is a vertex. In a road network, each intersection is a vertex. In a computer network, each device is a vertex. Vertices are connected by edges, which represent relationships between them.
Design a graph to model the layout of your school. Identify what the vertices and edges represent, and explain one question that	**Vertices could be rooms (classrooms, gym, cafeteria, office), edges could be hallways connecting them, and analyzing the graph could find the shortest path between any two rooms for evacuation planning.** — A school graph might have rooms as vertices and hallways/doors as edges. Edge weights could represent walking time. Useful questions include: What's the shortest evacuation route from each

analyzing this graph could answer.	classroom to an exit? Which hallway, if blocked, would disconnect the most rooms? How could adding one new hallway most improve traffic flow? This is real facility planning using graph theory!
Can a graph with 5 vertices all having odd degree exist?	**No, the sum of all degrees must be even** — See the correct answer above for the explanation.
A delivery company needs to visit 6 stores and return to the warehouse. Why is finding the shortest route through all stores so much harder than finding the shortest route between just two stores?	**With 6 stores, there are 720 possible orderings (6! = 720) to check, and the number grows factorially with more stops, making it far harder than a two-point shortest path.** — This is the Traveling Salesman Problem (TSP), one of the most famous problems in computer science. For 6 stores: 6! = 720 routes. For 20 stores: 20! ≈ 2.4 quintillion routes. Even the fastest computers can't check them all for large numbers. Two-point shortest path (like GPS) is solved efficiently by Dijkstra's algorithm, but TSP has no known fast solution — it's 'NP-hard.' Real delivery companies use approximation algorithms that find good (but not necessarily perfect) routes.
Why can a graph with 5 vertices and 11 edges not exist if the graph is simple (no self-loops or parallel edges)?	**A simple graph with 5 vertices can have at most 10 edges, calculated as 5 × 4 / 2 = 10, so 11 edges is impossible.** — In a simple graph (no duplicate edges, no self-loops), each pair of vertices can have at most one edge. With 5 vertices, the number of possible pairs is C(5,2) = 5×4/2 = 10. This maximum is called a complete graph (K_5). Since 11 > 10, such a graph cannot exist. This upper bound is fundamental in graph theory.
What is the degree of a vertex in a graph?	**The degree of a vertex is the number of edges connected to it.** — The degree of a vertex counts its connections. In a social network, a person's degree is their number of friends. A vertex with degree 0 (no connections) is called isolated. The Handshaking Lemma states that the sum of all vertex degrees equals twice the number of edges — because each edge contributes to the degree of two vertices.
A network has 5 computers connected by cables. One cable fails, disconnecting the network. An engineer says adding just one more cable would prevent any single cable failure from disconnecting the network. Is this always possible?	**Not always. If the original network is a tree (minimum connections), one extra cable creates one cycle, but there might still be bridges whose failure disconnects the network.** — It depends on the topology. A 5-vertex tree has 4 edges; adding a 5th creates exactly one cycle. Vertices in that cycle gain a backup path, but vertices outside it don't. To make every edge failure survivable, the graph needs to be '2-edge-connected,' meaning every pair of vertices has at least 2 edge-disjoint paths. This requires careful analysis, not just adding random cables.
Social media platforms show you 'People You May Know' suggestions. How might a graph algorithm find these suggestions by looking at friend-of-friend connections?	**The algorithm looks for people who share many mutual friends with you (vertices two edges away) but aren't directly connected to you yet.** — In the social graph, your direct friends are at distance 1. People at distance 2 (friends of friends) are candidates for 'People You May Know.' The algorithm counts how many mutual friends you share — more shared neighbors = stronger recommendation. This is essentially counting common neighbors in graph theory, one of the simplest but most powerful graph algorithms used in practice.
What does it mean for a graph to be 'connected'?	**A connected graph has a path between every pair of vertices — you can reach any vertex from any other vertex.** — A connected graph ensures reachability — from any vertex, there exists a path to every other vertex. If a graph is disconnected, it breaks into separate 'components' with no edges between them. The internet aims to be a connected graph so any computer can reach any other. Road networks in cities are typically connected.

A graph has vertices A, B, C, D, E with edges A-B, B-C, C-D, D-E, and E-A. What is the shortest path from A to C?	**The shortest path from A to C is A → B → C, which has length 2 (two edges).** — The graph forms a cycle A-B-C-D-E-A (a pentagon). From A to C, the clockwise path A→B→C has 2 edges, while the counterclockwise path A→E→D→C has 3 edges. The shortest path has length 2. There is no direct A-C edge, so length 1 is impossible. Finding shortest paths is one of the most important problems in graph theory.
What is an edge in graph theory?	**An edge is a connection between two vertices in a graph, representing a relationship or link.** — An edge connects two vertices, representing a relationship. In a road map graph, an edge is a road between two intersections. In a social network, an edge means two people are friends. Edges can be directed (one-way, like following someone on social media) or undirected (two-way, like a mutual friendship).
Two students debate: Student A says 'Graphs are just dots and lines — they can't represent anything complex.' Student B says 'Graphs can model almost any relationship.' Who makes the stronger argument?	**Student B is correct. Despite their simplicity, graphs model social networks, transportation systems, the internet, biological pathways, language relationships, and many more complex systems.** — Student B is right. The simplicity of graphs is their power — by abstracting complex systems into vertices and edges, we can apply the same mathematical tools to social networks, airline routes, disease spread, food webs, circuit design, and the internet. Google's PageRank, GPS navigation, social media recommendations, and epidemiological models all use graph algorithms.
Invent a board game that teaches graph theory concepts. Describe the game board, the rules, and which graph theory concept players would learn while playing.	**A good game might have players build a network of cities (vertices) and roads (edges), earning points for connecting all cities with minimum roads (spanning tree) while opponents can 'block' edges, teaching connectivity and tree concepts.** — A great graph theory game combines strategy with learning. Players could race to build spanning trees (connect all cities cheaply), find shortest paths (deliver goods efficiently), or create Eulerian circuits (visit all roads). Real games like Ticket to Ride, Pandemic, and Settlers of Catan all use graph theory — routes, connectivity, and network building are inherently fun because they mirror real decision-making.
In the famous Königsberg Bridge Problem, people tried to walk across all 7 bridges exactly once and return home. Euler proved this was impossible. What graph property did he use?	**Euler showed that for such a walk (Eulerian circuit) to exist, every vertex must have an even degree, but the Königsberg graph had vertices with odd degrees.** — Euler's breakthrough was realizing that for a circuit visiting every edge exactly once (Eulerian circuit), every vertex must have even degree. This is because each time you pass through a vertex, you use one edge to enter and one to leave — always in pairs. Königsberg had 4 vertices, all with odd degrees (3, 3, 3, 5), making the circuit impossible. This 1736 proof is considered the birth of graph theory!
What is a path in graph theory?	**A path is a sequence of vertices connected by edges where no vertex is visited more than once.** — A path visits vertices in sequence without repeating any vertex. The length of a path is the number of edges traversed. If a path starts and ends at the same vertex, it's called a cycle. Finding paths between vertices is crucial for GPS navigation, network routing, and solving mazes.
A student claims that if a graph has 6 vertices and each vertex has degree 2, the graph must be a single hexagon (6-cycle). Is this true?	**No, the graph could also be two separate triangles (3-cycles), since each vertex would still have degree 2.** — When every vertex has degree 2, the graph must consist of disjoint cycles (each vertex has one edge in and one edge out). With 6 vertices, the possibilities are: one 6-cycle (hexagon), one 3-cycle + one 3-cycle (two triangles), or one 4-cycle + isolated vertices with degree 2 (impossible without adjusting). The two valid options are the hexagon or two triangles. This shows that degree sequences don't uniquely determine a graph!
	A weighted graph assigns a numerical value to each edge, representing cost,

What is a weighted graph, and when would you use one?	**distance, time, or another quantity.** — In a weighted graph, each edge has a numerical weight. Road maps use distance weights for navigation. Airline routes use time or cost weights for booking. Network diagrams use bandwidth weights for routing data. Finding the shortest (lowest total weight) path in a weighted graph is one of the most practical problems in computer science — GPS devices solve this millions of times daily.
Why is a tree (in graph theory) always a connected graph with no cycles?	**A tree connects all vertices using the minimum number of edges (n-1 edges for n vertices) without forming any loops.** — A tree with n vertices has exactly n-1 edges — the minimum needed for connectivity. Adding any edge would create a cycle; removing any edge would disconnect it. Trees are everywhere in computer science: file systems (folders within folders), HTML document structure, tournament brackets, and decision trees. They're the most efficient way to connect all points.
What is the difference between a directed graph and an undirected graph?	**In a directed graph, edges have a direction (like one-way streets), while in an undirected graph, edges go both ways.** — In an undirected graph, friendship is mutual: if A is connected to B, then B is connected to A. In a directed graph (digraph), connections can be one-way: you can follow a celebrity on social media without them following you back. Road networks use directed graphs for one-way streets and undirected for two-way roads.

5. Combinatorics and Probability

✂ Question (fold →)	Answer
What is the complement of an event in probability?	**The complement of an event is everything that is NOT that event, and their probabilities add up to 1.** — The complement of event A (written A') includes all outcomes NOT in A. If P(A) = 0.3, then P(A') = 1 - 0.3 = 0.7. The probability of an event and its complement always sum to 1.
A pizza shop offers 3 sizes, 4 crust types, and 6 toppings. If you choose one of each, how many different pizza combinations are possible?	**There are 72 different combinations because 3 x 4 x 6 = 72.** — The Fundamental Counting Principle says to multiply the number of options at each step. 3 sizes x 4 crusts x 6 toppings = 72 total combinations.
What is the difference between a permutation and a combination?	**In permutations order matters, but in combinations order does not matter.** — Permutations count arrangements where order matters (ABC is different from BAC). Combinations count selections where order doesn't matter (choosing A, B, C is the same regardless of order).
A committee of 3 is chosen from 8 people. How many possible committees are there?	**There are 56 possible committees because C(8,3) = 8! / (3! x 5!) = 56.** — Committee members have no distinct roles, so order doesn't matter. C(8,3) = 8! / (3! x 5!) = (8 x 7 x 6) / (3 x 2 x 1) = 336/6 = 56.
How many different 4-digit PIN codes are possible using digits 0-9 if digits can be repeated?	**There are 10,000 possible PIN codes because 10 x 10 x 10 x 10 = 10,000.** — With repetition allowed, each of the 4 positions has 10 choices (0-9), and choices are independent. Total = 10^4 = 10,000 codes (from 0000 to 9999).
What is the probability of rolling a sum of 7 with two standard dice?	**The probability is 6/36 or 1/6, because there are 6 combinations that sum to 7 out of 36 total outcomes.** — Two dice produce 36 equally likely outcomes (6 x 6). The pairs that sum to 7 are: (1,6), (2,5), (3,4), (4,3), (5,2), (6,1) — that's 6 outcomes. So P(sum=7) = 6/36 = 1/6.
What is expected value, and how do you calculate it?	**Expected value is the average outcome you'd expect over many trials, calculated by multiplying each outcome by its probability and adding them together.** — Expected value (EV) is the long-run average. For a fair die: EV = (1)(1/6) + (2)(1/6) + (3)(1/6) + (4)(1/6) + (5)(1/6) + (6)(1/6) = 21/6 = 3.5. No single roll equals 3.5, but the average over many rolls approaches it.
A class has 10 students. How many ways can you choose a president, vice president, and secretary?	**There are 720 ways because 10 x 9 x 8 = 720.** — Since the positions are distinct (order matters) and one person can't hold multiple offices, this is a permutation: P(10,3) = 10 x 9 x 8 = 720.
What does it mean for two events to be independent in probability?	**Two events are independent when the outcome of one event does not affect the probability of the other event.** — Independent events don't influence each other. For example, flipping a coin (heads/tails) doesn't change the probability of rolling a 6 on a die. Mathematically, P(A and B) = P(A) x P(B) for independent events.
In a game, you spin a spinner divided into 5 equal sections (red, blue, green, yellow, purple) and flip a coin. What is	**The probability is 1/10 because (1/5) x (1/2) = 1/10.** — The spinner and coin flip are independent events. P(blue) = 1/5 and P(heads) = 1/2. For independent events,

the probability of landing on blue AND getting heads?	P(A and B) = P(A) x P(B) = 1/5 x 1/2 = 1/10.
A bag contains 5 red and 3 blue marbles. You draw 2 marbles without replacement. What is the probability that you get one of each color?	**The probability is 15/28, found by calculating: (5/8)(3/7) + (3/8)(5/7) = 15/56 + 15/56 = 30/56 = 15/28.** — Two paths: (Red then Blue) = (5/8)(3/7) = 15/56, and (Blue then Red) = (3/8)(5/7) = 15/56. P(one of each) = 15/56 + 15/56 = 30/56 = 15/28 ≈ 53.6%.
You flip a fair coin 3 times. What is the probability of getting exactly 2 heads?	**The probability is 3/8 because there are 3 favorable outcomes (HHT, HTH, THH) out of 8 total outcomes.** — All outcomes: HHH, HHT, HTH, THH, HTT, THT, TTH, TTT (8 total). Outcomes with exactly 2 heads: HHT, HTH, THH (3 favorable). Probability = 3/8.
A bag has 3 red marbles, 4 blue marbles, and 5 green marbles. If you draw one marble without looking, what is the probability of drawing a blue marble?	**The probability is 4/12 or 1/3, because there are 4 blue marbles out of 12 total.** — Total marbles = 3 + 4 + 5 = 12. Blue marbles = 4. Probability = favorable/total = 4/12 = 1/3.
Design a carnival game that is fair (expected value of $0 for the player). The game uses a spinner with 4 equal sections.	**One fair design: charge $1 to play, one section wins $4, three sections win $0. EV = (1/4)($4) - $1 = $1 - $1 = $0.** — For a fair game, EV must equal $0. With 4 equal sections and a $1 entry fee: if 1 section wins $4 and 3 win $0, then EV = (1/4)($4) + (3/4)($0) - $1 = $1 - $1 = $0. Players break even on average.
A game costs $2 to play. You roll a die: if you roll a 6, you win $10; otherwise you win nothing. Should you play this game?	**The expected value is -$1/3 per game (lose about 33 cents on average), so mathematically you should not play.** — EV = P(win) x prize - cost = (1/6)($10) - $2 = $10/6 - $2 = $1.67 - $2 = -$0.33. On average you lose about 33 cents per game. Over many plays, you'd lose money, so it's not a fair game.
How many 3-letter codes can be made from the letters A, B, C, D, E if no letter can be repeated?	**There are 60 codes because 5 x 4 x 3 = 60.** — Since order matters (ABC is different from BAC) and no repetition is allowed, this is P(5,3) = 5 x 4 x 3 = 60. The first position has 5 choices, the second has 4, and the third has 3.
How many different ways can you distribute 3 identical stickers among 4 friends (each friend can get 0 or more stickers)?	**There are 20 ways, using the stars and bars method: C(3+4-1, 4-1) = C(6,3) = 20.** — Distributing n identical items among k groups uses the 'stars and bars' formula: C(n+k-1, k-1). Here, C(3+4-1, 4-1) = C(6,3) = 20. The distributions include (3,0,0,0), (2,1,0,0), (1,1,1,0), and all their rearrangements.
In a class of 25 students, what is the approximate probability that at least two students share a birthday? (Assume 365 days in a year.)	**The probability is about 57%, which is surprisingly high despite having far fewer students than days.** — This is the Birthday Problem. It's easier to calculate the complement: P(no shared birthdays) = (365/365) x (364/365) x ... x (341/365). This gives about 0.43, so P(at least one shared birthday) ≈ 1 - 0.43 = 0.57 or 57%. The high probability surprises most people!
How many ways can you arrange 4 different books on a shelf?	**There are 24 ways to arrange 4 books, since 4! = 4 x 3 x 2 x 1 = 24.** — Each position reduces the number of available books by one. The first slot has 4 choices, the second has 3, the third has 2, and the last has 1. Multiplying gives 4! = 24.
You draw two cards from a standard 52-card deck without	**The probability is (13/52) x (12/51) = 156/2652 = 1/17.** — The first card has a

replacement. What is the probability that both are hearts?	13/52 chance of being a heart. If it is, there are now 12 hearts left in 51 remaining cards. P(both hearts) = (13/52) x (12/51) = 156/2652 = 1/17.
A game has a 1/6 chance of winning each round. What is the probability of losing 3 rounds in a row?	**The probability is (5/6)^3 = 125/216, which is about 57.9%.** — P(lose one round) = 1 - 1/6 = 5/6. Since rounds are independent, P(lose 3 in a row) = (5/6) x (5/6) x (5/6) = 125/216 ≈ 0.579 or about 57.9%.
Create a word problem that requires using both permutations and combinations to solve.	**Example: A teacher picks a 3-person team from 8 students (C(8,3) = 56), then assigns captain, co-captain, and recorder roles (3! = 6). Total arrangements = 56 x 6 = 336.** — Multi-step problems often combine both. First, choosing the team is a combination (order doesn't matter: C(8,3) = 56 ways). Then assigning distinct roles is a permutation (order matters: 3! = 6 ways). By the multiplication principle, total = 56 x 6 = 336.
You have 5 friends and want to invite exactly 2 to a sleepover. How many different pairs can you choose?	**You can choose 10 different pairs, calculated as 5! / (2! x 3!) = 10.** — Since the order of choosing friends doesn't matter, this is a combination. C(5,2) = 5! / (2! x 3!) = (5 x 4) / (2 x 1) = 10 different pairs.
Why is P(A or B) = P(A) + P(B) - P(A and B) instead of just P(A) + P(B)?	**We subtract P(A and B) to avoid counting the overlap where both events happen, which would otherwise be counted twice.** — When events overlap (share outcomes), adding P(A) + P(B) counts the shared outcomes twice. Subtracting P(A and B) corrects this double-counting. This is the inclusion-exclusion principle from set theory applied to probability.
What is Pascal's Triangle, and how does it relate to combinations?	**Pascal's Triangle is a triangular array where each number is the sum of the two numbers above it, and the entries represent combination values C(n,r).** — Pascal's Triangle starts with 1 at the top. Each entry is the sum of the two entries above it. Row n (starting from 0) contains the combination values C(n,0) through C(n,n). For example, row 4 is 1,4,6,4,1 which gives C(4,0) through C(4,4).

6. Recursion and Sequences

✂ Question (fold →)	Answer
What is the golden ratio, and how does it relate to the Fibonacci sequence?	**The golden ratio (approximately 1.618) is the value that the ratio of consecutive Fibonacci numbers approaches as the numbers get larger.** — As Fibonacci numbers grow, the ratio F(n+1)/F(n) approaches φ ≈ 1.61803... The ratios are: 2/1=2, 3/2=1.5, 5/3≈1.667, 8/5=1.6, 13/8=1.625, 21/13≈1.615... converging to φ. The golden ratio appears in art, architecture, and nature.
A recursive sequence is defined as: a(1) = 1, a(2) = 1, a(n) = a(n-1) + 2 x a(n-2). Find a(5).	**a(5) = 11. Working out: a(3) = 1 + 2(1) = 3, a(4) = 3 + 2(1) = 5, a(5) = 5 + 2(3) = 11.** — a(1) = 1, a(2) = 1. a(3) = a(2) + 2 x a(1) = 1 + 2(1) = 3. a(4) = a(3) + 2 x a(2) = 3 + 2(1) = 5. a(5) = a(4) + 2 x a(3) = 5 + 2(3) = 5 + 6 = 11.
The Collatz conjecture says: if n is even, divide by 2; if n is odd, multiply by 3 and add 1. Starting from 6, trace the sequence until it reaches 1.	**Starting from 6: 6, 3, 10, 5, 16, 8, 4, 2, 1. It takes 8 steps to reach 1.** — 6 (even→3), 3 (odd→10), 10 (even→5), 5 (odd→16), 16 (even→8), 8 (even→4), 4 (even→2), 2 (even→1). The Collatz conjecture (still unproven!) states that every positive integer eventually reaches 1 through this process.
What is the base case in the recursive definition of n! (n factorial)?	**0! = 1** — See the correct answer above for the explanation.
Design a recursive sequence inspired by a real-world scenario (like population growth, compound interest, or a game score). Define the base case and recursive rule.	**Example — Savings account: S(0) = $100 (initial deposit), S(n) = 1.05 x S(n-1) + 10 (5% interest plus $10 monthly deposit). This models real compound growth with regular contributions.** — Many real-world processes are naturally recursive. The savings example: S(0)=$100, S(n)=1.05 x S(n-1) + 10 captures both interest growth (multiplicative) and regular deposits (additive). Other examples: bacteria doubling, loan payments, game levels, or viral sharing on social media.
The Fibonacci sequence starts 1, 1, 2, 3, 5, 8, 13, ... What are the next two numbers?	**The next two numbers are 21 and 34, because each number is the sum of the previous two.** — In the Fibonacci sequence, F(n) = F(n-1) + F(n-2). So the next number is 8 + 13 = 21, and the one after is 13 + 21 = 34. The sequence continues: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
The Tower of Hanoi puzzle with n disks requires a minimum of 2^n - 1 moves. How many moves does a 6-disk puzzle require?	**A 6-disk Tower of Hanoi requires 63 moves, because 2^6 - 1 = 64 - 1 = 63.** — The Tower of Hanoi formula is T(n) = 2^n - 1. For 6 disks: T(6) = 2^6 - 1 = 64 - 1 = 63. This is also recursive: T(n) = 2 x T(n-1) + 1, with T(1) = 1. You move n-1 disks aside (T(n-1) moves), move the largest disk (1 move), then move n-1 disks back (T(n-1) moves).
Use the domino analogy to explain why both parts of mathematical induction are necessary.	**The base case is like pushing the first domino. The inductive step ensures each domino knocks down the next. Without either part, the chain fails.** — Without the base case (pushing the first domino), nothing starts falling even if dominoes are perfectly arranged. Without the inductive step (dominoes correctly spaced), pushing the first domino might only knock down a few. You need both: a starting point AND a guarantee that each case triggers the next.
What is the difference	**The recursive formula is a(1) = 2, a(n) = 2 x a(n-1). The explicit formula is a(n) =**

between a recursive formula and an explicit (closed-form) formula for the sequence 2, 4, 8, 16, 32, ...?	**2^n. Both define the same sequence differently.** — The recursive formula a(n) = 2 x a(n-1) tells you each term from the previous one. The explicit formula a(n) = 2^n lets you compute any term directly. For the 10th term: recursive requires computing terms 1-9 first, but explicit gives 2^10 = 1024 immediately.
Trace through the recursive calculation: fib(5), where fib(1) = 1, fib(2) = 1, and fib(n) = fib(n-1) + fib(n-2).	**fib(5) = fib(4) + fib(3) = (fib(3) + fib(2)) + (fib(2) + fib(1)) = ((fib(2) + fib(1)) + 1) + (1 + 1) = ((1 + 1) + 1) + 2 = 5.** — Tracing: fib(5) = fib(4) + fib(3). Then fib(4) = fib(3) + fib(2) = fib(3) + 1. And fib(3) = fib(2) + fib(1) = 1 + 1 = 2. So fib(4) = 2 + 1 = 3. Finally, fib(5) = 3 + 2 = 5.
Create a recursive rule for a sequence where each term is the average of the two preceding terms, starting with a(1) = 0 and a(2) = 10. Find the first 6 terms.	**Rule: a(n) = (a(n-1) + a(n-2)) / 2. Terms: 0, 10, 5, 7.5, 6.25, 6.875. The sequence converges toward 6.667 (which is 20/3).** — a(1)=0, a(2)=10, a(3)=(10+0)/2=5, a(4)=(5+10)/2=7.5, a(5)=(7.5+5)/2=6.25, a(6)=(6.25+7.5)/2=6.875. The sequence oscillates and converges to the weighted average 2/3 x 10 = 6.667... This shows how recursive averaging creates convergent behavior.
What does it mean for something to be defined recursively?	**A recursive definition defines something in terms of itself, with a base case that stops the self-reference.** — Recursion defines something using a simpler version of itself. It needs two parts: a base case (the simplest case with a direct answer) and a recursive case (which reduces the problem to a simpler version). Without a base case, the definition would loop forever.
What is the closed-form (explicit) formula for the arithmetic sequence 3, 7, 11, 15, 19, ...?	**The explicit formula is a(n) = 4n - 1, where n starts at 1.** — The first term a(1) = 3 and the common difference d = 4. Using the formula a(n) = a(1) + (n-1)d = 3 + (n-1)(4) = 3 + 4n - 4 = 4n - 1. Verify: a(1) = 4(1) - 1 = 3, a(2) = 4(2) - 1 = 7, a(3) = 4(3) - 1 = 11.
What is mathematical induction, and why is it useful?	**Mathematical induction is a proof technique that proves a statement is true for all natural numbers by showing it holds for a base case and that if it holds for n, it also holds for n+1.** — Induction has two steps: (1) Base case: prove the statement for n=1 (or the smallest case). (2) Inductive step: assume it's true for some n (inductive hypothesis), then prove it for n+1. Like dominoes, this chain reaction proves it for all natural numbers.
Why is a recursive definition sometimes preferred over a closed-form formula?	**Recursive definitions are sometimes simpler to understand and write, especially when each step depends directly on the previous step.** — Recursive definitions model step-by-step processes naturally. The Fibonacci recursive rule (add the last two numbers) is easy to understand, while its closed-form (Binet's formula) involves the golden ratio and square root of 5. However, closed-form formulas are faster for computing the nth term directly.
Define the sequence T(n) recursively: T(1) = 3, T(n) = 2 x T(n-1) + 1. Find T(4).	**T(4) = 31, found by computing T(2) = 7, T(3) = 15, T(4) = 2 x 15 + 1 = 31.** — T(1) = 3. T(2) = 2 x 3 + 1 = 7. T(3) = 2 x 7 + 1 = 15. T(4) = 2 x 15 + 1 = 31. Each step doubles the previous value and adds 1. The sequence is 3, 7, 15, 31, 63, ... which equals 2^(n+1) - 1.
What are the Lucas numbers, and how are they similar to and different from the Fibonacci numbers?	**Lucas numbers follow the same recursive rule as Fibonacci (add the previous two) but start with different base cases: L(1) = 2, L(2) = 1, giving 2, 1, 3, 4, 7, 11, 18, ...** — Lucas numbers: 2, 1, 3, 4, 7, 11, 18, 29, 47, ... Same rule as Fibonacci (L(n) = L(n-1) + L(n-2)) but different starting values (2 and 1 instead of 1 and 1). This shows how base cases dramatically change a recursive sequence's values while maintaining the same pattern.

Compare the efficiency of computing fib(20) recursively versus iteratively. Which is faster and why?	**Iterative is much faster because naive recursion recalculates the same values many times, while iteration computes each value once from left to right.** — Naive recursion computes fib(20) by calling fib(19) and fib(18), which both call fib(17), and so on. This creates an exponential explosion — fib(3) is computed 2,584 times! Iteration computes each Fibonacci number once in a loop, requiring only 20 steps. Memoization (caching results) can fix the recursive version.
A cell divides into 2 cells every hour. Starting with 1 cell, write a recursive formula and find how many cells exist after 8 hours.	**Recursive formula: C(0) = 1, C(n) = 2 x C(n-1). After 8 hours: C(8) = 2^8 = 256 cells.** — Base case: C(0) = 1 cell at the start. Recursive case: C(n) = 2 x C(n-1) because each cell divides into 2. C(1)=2, C(2)=4, C(3)=8, C(4)=16, C(5)=32, C(6)=64, C(7)=128, C(8)=256.
How many ancestors do you have going back 10 generations (parents, grandparents, great-grandparents, etc.)?	**You have 2 + 4 + 8 + ... + 1024 = 2046 ancestors over 10 generations, which equals 2^11 - 2.** — Each generation doubles: 2, 4, 8, 16, ... 2^10 = 1024. The total is 2 + 4 + 8 + ... + 1024 = 2(2^10 - 1)/(2-1) = 2(1023) = 2046. This is a geometric series sum. (In reality, ancestors overlap due to intermarriage.)
What are the base case and recursive case for the factorial function n!?	**The base case is 0! = 1, and the recursive case is n! = n x (n-1)! for n > 0.** — The factorial function has base case 0! = 1 (by definition). The recursive case says n! = n x (n-1)!. So 5! = 5 x 4! = 5 x 4 x 3! = 5 x 4 x 3 x 2 x 1! = 5 x 4 x 3 x 2 x 1 x 0! = 120.
Write a recursive rule for the sequence: 2, 6, 18, 54, 162, ...	**The recursive rule is a(1) = 2 and a(n) = 3 x a(n-1), because each term is 3 times the previous term.** — Each term is 3 times the previous term (6/2=3, 18/6=3, 54/18=3). The recursive definition is: a(1) = 2, a(n) = 3 x a(n-1) for n > 1. This is a geometric sequence with first term 2 and common ratio 3.
The Sierpinski Triangle starts with one triangle and recursively divides it. After 4 iterations, how many small triangles are shaded?	**After 4 iterations, there are 3^4 = 81 shaded triangles, because each iteration triples the number of shaded triangles.** — The Sierpinski Triangle is a fractal. At each iteration, each shaded triangle becomes 3 smaller shaded triangles. Starting with 1: after step 1 = 3, step 2 = 9, step 3 = 27, step 4 = 81. The formula is 3^n after n iterations.
What is the sum of the first n natural numbers? (1 + 2 + 3 + ... + n)	**The sum is n(n+1)/2. For example, 1 + 2 + 3 + ... + 10 = 10(11)/2 = 55.** — Young Gauss discovered this by pairing: 1+n, 2+(n-1), 3+(n-2), etc. Each pair sums to (n+1), and there are n/2 pairs. Total = n(n+1)/2. This is also the nth triangular number. For n=100: 100(101)/2 = 5050.
What happens if a recursive definition has no base case?	**Without a base case, the recursion never stops — it creates an infinite loop that keeps calling itself forever.** — A base case is essential for recursion to terminate. Without it, the function keeps calling itself with no way to stop: f(5) calls f(4), which calls f(3), which calls f(2)... forever. In computing, this crashes the program with a 'stack overflow' error.

7. Number Theory and Cryptography

✂ Question (fold →)	Answer
What is the Chinese Remainder Theorem, explained simply?	**The Chinese Remainder Theorem says if you know the remainders of a number when divided by several coprime divisors, you can uniquely determine the remainder when divided by their product.** — If you know n mod 3 = 2 and n mod 5 = 3, then n mod 15 = 8 (since 8 mod 3 = 2 and 8 mod 5 = 3). The CRT guarantees a unique solution mod (3 x 5) = 15 when the divisors are coprime. This 1,500+ year-old theorem is used in RSA computation and error-correcting codes.
Find the greatest common divisor (GCD) of 48 and 36 using the Euclidean algorithm.	**GCD(48, 36) = 12. Steps: 48 = 1 x 36 + 12, then 36 = 3 x 12 + 0. When the remainder is 0, the last non-zero remainder (12) is the GCD.** — Euclidean algorithm: GCD(48,36) → 48 = 1(36) + 12, so GCD(36,12) → 36 = 3(12) + 0. Remainder is 0, so GCD = 12. The algorithm works because GCD(a,b) = GCD(b, a mod b), and it always terminates because remainders decrease.
What is a one-way function, and why is it important for cryptography?	**A one-way function is easy to compute in one direction but extremely difficult to reverse. It's the basis of public-key cryptography — anyone can encrypt, but only the key holder can decrypt.** — One-way functions are computational trapdoors: f(x) is quick to compute, but finding x from f(x) is impractical. Prime multiplication is one-way: 317 x 331 = 104,927 is instant, but factoring 104,927 back into 317 x 331 takes much longer. This asymmetry enables RSA and other public-key systems.
Design a simple substitution cipher that is harder to break than the Caesar cipher. Explain why it's more secure.	**Use a random mapping where each letter maps to a different letter (not just a fixed shift). This has 26! (about 4 x 10^26) possible keys instead of just 25, making brute force impractical.** — A general substitution cipher assigns each letter to a random different letter: A→M, B→X, C→F, etc. There are 26! ≈ 4 x 10^26 possible mappings, making brute force impossible. However, frequency analysis (E is most common in English) can still crack it. This is why modern ciphers use much more sophisticated methods.
Create a simple number-based encryption scheme using modular arithmetic. Encrypt the number 15 using your scheme and show how to decrypt it.	**Example scheme: Encrypt: C = (M x 3 + 7) mod 26. Encrypt 15: C = (15 x 3 + 7) mod 26 = 52 mod 26 = 0. Decrypt: find M where (M x 3 + 7) mod 26 = 0, giving M = 15.** — Encrypt: C = (M x k + s) mod n, where k is the multiplier and s is the shift. Decrypt: M = ((C - s) x k_inv) mod n, where k_inv is the modular inverse of k. For k=3, n=26: k_inv = 9 (because 3 x 9 = 27 ≡ 1 mod 26). Decrypt: M = ((0 - 7) x 9) mod 26 = (-63) mod 26 = 15.
What is Euler's totient function ϕ(n), and what is ϕ(12)?	**ϕ(n) counts the numbers from 1 to n that are coprime to n (share no common factors other than 1). ϕ(12) = 4, because 1, 5, 7, and 11 are coprime to 12.** — ϕ(12) counts integers from 1 to 12 that share no common factor with 12 (GCD = 1). Numbers 2,3,4,6,8,9,10,12 share a factor with 12. The coprime numbers are 1, 5, 7, 11 → ϕ(12) = 4. Euler's totient is central to RSA encryption's key generation.
What is 23 mod 7? What is 100 mod 12?	**23 mod 7 = 2 (because 23 = 3 x 7 + 2), and 100 mod 12 = 4 (because 100 = 8 x 12 + 4).** — 23 ÷ 7 = 3 with remainder 2, so 23 mod 7 = 2. 100 ÷ 12 = 8 with remainder 4, so 100 mod 12 = 4. The mod operation always gives a result between 0 and (divisor - 1).
How does a hash function work, and	**A hash function converts any input into a fixed-size output (hash). It's a one-way function: many different inputs can produce the same hash, so you can't determine the original input from the hash alone.** — Hash functions map inputs of any size to a

why can't you reverse it?	fixed-size output. They're one-way because information is lost: many inputs map to the same hash (collision). Your password is stored as a hash — when you log in, the system hashes your input and compares hashes, never storing or seeing your actual password.
What is Fermat's Little Theorem, and how is it used in cryptography?	**Fermat's Little Theorem states that if p is prime and a is not divisible by p, then a^(p-1) mod p = 1. It's used in RSA to efficiently compute modular exponentiation.** — Fermat's Little Theorem: if p is prime and GCD(a,p) = 1, then a^(p-1) ≡ 1 (mod p). Example: 3^4 = 81, and 81 mod 5 = 1. In RSA, this theorem ensures that encryption followed by decryption returns the original message. It also enables efficient primality testing.
What is modular arithmetic, and what does '17 mod 5' equal?	**Modular arithmetic is the math of remainders after division. 17 mod 5 = 2, because 17 divided by 5 is 3 remainder 2.** — Modular arithmetic finds the remainder after division. 17 ÷ 5 = 3 with remainder 2, so 17 mod 5 = 2. Clock arithmetic is modular: 17 hours on a 12-hour clock shows 5 (17 mod 12 = 5). Modular arithmetic is fundamental to cryptography.
What is the Fundamental Theorem of Arithmetic?	**Every integer greater than 1 can be expressed as a unique product of prime numbers, regardless of the order of factors.** — The Fundamental Theorem of Arithmetic states that every integer > 1 has exactly one prime factorization (ignoring order). For example, 84 = 2 x 2 x 3 x 7 = 2^2 x 3 x 7, and there is no other way to express 84 as a product of primes. This uniqueness is the foundation of modern cryptography.
What is the Sieve of Eratosthenes, and how does it find prime numbers?	**The Sieve of Eratosthenes finds all primes up to a given number by systematically crossing out multiples of each prime, starting with 2.** — List numbers from 2 to N. Start with 2: it's prime, so cross out 4, 6, 8, 10... Next uncrossed number is 3: cross out 9, 15, 21... Next is 5: cross out 25, 35... The remaining uncrossed numbers are all prime. This 2,000+ year-old algorithm is still used in modern computing.
Using the Sieve of Eratosthenes, find all prime numbers between 1 and 30.	**The primes between 1 and 30 are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 — there are 10 primes.** — Start: 2-30. Cross out multiples of 2 (4,6,8,...,30). Cross out multiples of 3 (9,15,21,27). Cross out multiples of 5 (25). Stop since 7^2 = 49 > 30. Remaining: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. Note: 1 is neither prime nor composite by definition.
Why is the Caesar cipher considered weak by modern standards?	**It is weak because there are only 25 possible shifts, so an attacker can try all of them quickly (brute force attack).** — With only 25 possible shifts (1 through 25), a brute-force attack simply tries each one. Additionally, letter frequency analysis can crack it: 'E' is the most common English letter, so the most frequent letter in the ciphertext likely represents 'E.' Modern ciphers use keys with trillions of possibilities.
Two numbers are 'congruent modulo n' if they have the same remainder when divided by n. Are 38 and 14 congruent modulo 8?	**Yes, 38 and 14 are congruent modulo 8 because 38 mod 8 = 6 and 14 mod 8 = 6 — they have the same remainder.** — 38 mod 8 = 6 (since 38 = 4 x 8 + 6) and 14 mod 8 = 6 (since 14 = 1 x 8 + 6). Same remainder, so they are congruent modulo 8. Written: 38 ≡ 14 (mod 8). Equivalently, 38 - 14 = 24, which is divisible by 8.
What is a Caesar cipher, and how would you encrypt the word 'MATH' with a shift of 3?	**A Caesar cipher shifts each letter by a fixed number. With shift 3, MATH becomes PDWK (M→P, A→D, T→W, H→K).** — The Caesar cipher replaces each letter with the one a fixed number of positions later in the alphabet. With shift 3: M→P, A→D, T→W, H→K. Julius Caesar used this to communicate with his generals. It's easily broken because there are only 25 possible shifts to try.
What does it mean for two numbers to be 'coprime'?	**Their greatest common divisor is 1** — See the correct answer above for the explanation.

What is the difference between symmetric and asymmetric encryption?	**Symmetric encryption uses the same key to encrypt and decrypt, while asymmetric encryption uses a public key to encrypt and a different private key to decrypt.** — Symmetric encryption (like AES) uses one shared secret key. It's fast but requires securely sharing the key. Asymmetric encryption (like RSA) uses a public key anyone can use to encrypt, but only the holder of the matching private key can decrypt. RSA's security relies on the difficulty of factoring large numbers.
What are twin primes? Give three examples of twin prime pairs.	**Twin primes are pairs of primes that differ by exactly 2. Examples: (3, 5), (11, 13), and (17, 19).** — Twin primes are pairs of primes with a gap of 2. Examples: (3,5), (5,7), (11,13), (17,19), (29,31), (41,43). The Twin Prime Conjecture says there are infinitely many such pairs, but this remains unproven — one of math's great unsolved problems!
Explain why there are infinitely many prime numbers, using Euclid's argument.	**Assume there are finitely many primes. Multiply them all together and add 1. This new number isn't divisible by any known prime (remainder 1), so it must have a new prime factor — a contradiction.** — Euclid's proof (300 BCE): Suppose there are finitely many primes: p1, p2, ..., pk. Form N = p1 x p2 x ... x pk + 1. N gives remainder 1 when divided by any of our primes, so none of them divide N. But N must have a prime factor (Fundamental Theorem), and it's not in our list. Contradiction! So there must be infinitely many primes.
In modular arithmetic, what is (7 + 9) mod 5?	**(7 + 9) mod 5 = 16 mod 5 = 1, because 16 divided by 5 gives remainder 1.** — Two methods: (1) Direct: 7 + 9 = 16, and 16 mod 5 = 1. (2) Reduce first: 7 mod 5 = 2, 9 mod 5 = 4, then (2 + 4) mod 5 = 6 mod 5 = 1. Both give the same answer. Note: the first distractor has the right idea but got 3 instead of 1 (arithmetic error demonstration).
Why are prime numbers important for modern internet security?	**Large prime numbers are the foundation of RSA encryption because multiplying two large primes is easy, but finding the original primes from their product is extremely difficult.** — RSA encryption relies on the asymmetry: multiplying two primes (like 7919 x 7907 = 62,607,133) is quick, but factoring the product back into primes is extremely slow for large numbers. RSA uses primes with hundreds of digits, where factoring would take millions of years even with modern computers.
What does it mean for two numbers to be coprime (relatively prime)?	**Two numbers are coprime if their greatest common divisor (GCD) is 1, meaning they share no common factors other than 1.** — Coprime (or relatively prime) numbers share no common factor except 1. Examples: 8 and 15 (factors of 8: 1,2,4,8; factors of 15: 1,3,5,15 — only 1 in common). Note: the numbers themselves don't need to be prime. Even 4 and 9 are coprime (GCD = 1).
Decrypt the message 'KHOOR' using a Caesar cipher with shift 3.	**KHOOR decrypts to HELLO by shifting each letter back by 3: K→H, H→E, O→L, O→L, R→O.** — Decryption reverses the encryption: shift each letter back by 3. K(11)→H(8), H(8)→E(5), O(15)→L(12), O(15)→L(12), R(18)→O(15). KHOOR → HELLO. Caesar cipher encryption and decryption are inverse operations.
If you add any two even numbers, the result is always even. Use divisibility to explain why.	**An even number can be written as 2a and another as 2b. Their sum is 2a + 2b = 2(a + b), which is 2 times an integer, so it's always even.** — Any even number can be written as 2k for some integer k. If we have 2a and 2b, then 2a + 2b = 2(a+b). Since (a+b) is an integer, 2(a+b) is divisible by 2, making it even. This is a direct proof using the definition of even numbers.

8. Algorithm Design and Complexity

✂ Question (fold →)	Answer
How does binary search work, and what is its time complexity?	**Binary search works on a sorted list by repeatedly checking the middle element and eliminating half the remaining elements. Its time complexity is O(log n).** — Binary search: (1) Check middle element. (2) If it matches, done. (3) If target is smaller, search the left half. (4) If larger, search the right half. Each step halves the search space: 1000 items → 500 → 250 → 125 → ... → 1 in about 10 steps. $\log_2(1000) \approx 10$, so O(log n).
What is the divide and conquer strategy in algorithm design?	**Divide and conquer breaks a problem into smaller subproblems, solves each one independently, then combines the solutions to solve the original problem.** — Divide and conquer has three steps: (1) Divide the problem into smaller subproblems. (2) Conquer (solve) each subproblem recursively. (3) Combine the solutions. Merge sort divides a list into halves, sorts each half (recursively), then merges the sorted halves. This gives O(n log n) instead of $O(n^2)$.
Rank these Big O complexities from fastest to slowest: $O(n^2)$, O(1), O(n log n), O(n), O(2^n).	**From fastest to slowest: O(1), O(n), O(n log n), $O(n^2)$, O(2^n).** — For n=100: O(1)=1, O(n)=100, O(n log n)≈664, $O(n^2)$=10,000, O(2^n)≈1.27×10^30. The gap between polynomial and exponential is dramatic: $O(n^2)$ is manageable for millions of items, but O(2^n) becomes impractical beyond about 30-40 items.
You need to sort 5 cards: [8, 3, 1, 5, 2]. Show the steps using selection sort.	**Selection sort finds the minimum and places it at the front. Steps: [1,3,8,5,2] → [1,2,8,5,3] → [1,2,3,5,8] → [1,2,3,5,8] → done.** — Selection sort: Pass 1: min of [8,3,1,5,2] is 1 at index 2, swap with index 0 → [1,3,8,5,2]. Pass 2: min of [3,8,5,2] is 2 at index 4, swap with index 1 → [1,2,8,5,3]. Pass 3: min of [8,5,3] is 3 at index 4, swap with index 2 → [1,2,3,5,8]. Pass 4: min of [5,8] is 5, already in place. Done!
You have a sorted list of 1024 names. Using binary search, what is the maximum number of comparisons needed to find any name?	**The maximum is 10 comparisons, because $\log_2(1024) = 10$. Each comparison halves the remaining possibilities.** — Binary search halves the search space with each comparison. Starting with 1024 names: after 1 comparison = 512 remaining, after 2 = 256, ... after 10 = 1. So at most 10 comparisons. In general, binary search on n items needs at most $\lceil \log_2(n) \rceil$ comparisons. For 1024 = 2^10, that's exactly 10.
Design an efficient algorithm to determine if a word is a palindrome (reads the same forwards and backwards). What is its time complexity?	**Compare the first character with the last, second with second-to-last, and so on, moving inward. Stop when the pointers meet in the middle. Time complexity is O(n/2) = O(n).** — Two-pointer approach: set left = 0 (start), right = n-1 (end). While left < right: if word[left] ≠ word[right], return false. Otherwise, left++, right--. If all pairs match, return true. This is O(n) time, O(1) extra space. Alternative: reverse and compare is also O(n) but creates a copy (O(n) space).
What is an algorithm?	**An algorithm is a step-by-step set of instructions for solving a problem or completing a task, where each step is clear and unambiguous.** — An algorithm is a finite sequence of well-defined instructions for solving a class of problems. Algorithms predate computers: Euclid's algorithm (300 BCE) for finding the GCD is one of the oldest. Key properties: finite, definite (unambiguous), effective (each step can be done), and produces output.

Compare three approaches to computing the sum 1 + 2 + 3 + ... + n: a loop, recursion, and a formula. Which is most efficient?	**The formula n(n+1)/2 is O(1) — instant regardless of n. A loop is O(n) — adds one number at a time. Recursion is also O(n) but uses O(n) stack space. The formula is most efficient.** — Formula: n(n+1)/2 — O(1) time, O(1) space. One multiplication and one division regardless of n. Loop: sum += i for i=1..n — O(n) time, O(1) space. Recursion: f(n) = n + f(n-1) — O(n) time, O(n) space (call stack). Gauss's formula is clearly superior, demonstrating that mathematical insight can dramatically outperform algorithmic brute force.
What is memoization, and how does it improve the recursive Fibonacci algorithm?	**Memoization stores previously computed results to avoid redundant calculations. It improves recursive Fibonacci from O(2^n) to O(n) by caching each Fibonacci value after computing it once.** — Naive recursive Fibonacci recomputes the same values exponentially: fib(50) would take trillions of calls. Memoization stores each result: when fib(k) is first computed, save it. If fib(k) is needed again, return the cached value in O(1). This reduces fib(n) from O(2^n) to O(n) — a massive improvement.
What is the difference between a brute force approach and an optimized algorithm for finding if a number n is prime?	**Brute force checks all numbers from 2 to n-1 (O(n) checks). An optimized approach only checks up to the square root of n (O($\sqrt{n}$) checks), because if n has a factor larger than $\sqrt{n}$, it must also have one smaller than $\sqrt{n}$.** — Brute force: test divisibility by 2, 3, 4, ..., n-1. For n=1,000,000, that's ~1 million checks. Optimized: only test up to $\sqrt{n} \approx 1000$, since if n = a × b, one factor must be ≤ $\sqrt{n}$. Further optimization: skip even numbers after 2, reducing checks to ~500. For 1,000,000 this is 1000x faster!
Why is merge sort (O(n log n)) faster than bubble sort (O(n^2)) for large lists?	**For large n, n log n grows much slower than n^2. With 1 million items, merge sort does about 20 million operations versus bubble sort's 1 trillion operations.** — For n = 1,000,000: $n^2 = 10^{12}$ (1 trillion), while $n \log_2 n \approx 20{,}000{,}000$ (20 million). Merge sort is about 50,000 times faster! The O(n log n) vs O(n^2) gap widens with larger inputs. For small lists (n < 50), bubble sort may actually be faster due to lower overhead.
An algorithm takes 2 seconds for 100 items. If it's O(n^2), approximately how long will it take for 1000 items?	**It will take approximately 200 seconds, because when n increases 10x with O(n^2), the time increases 100x ($10^2 = 100$).** — With O(n^2): time $\propto n^2$. For n=100: 2 seconds = $k \times 100^2 = k \times 10{,}000$, so k = 0.0002. For n=1000: time = $0.0002 \times 1000^2 = 0.0002 \times 1{,}000{,}000 = 200$ seconds. Input grew 10x, so time grew 10^2 = 100x. This is why O(n^2) algorithms struggle with large inputs.
A linear search checks 1,000,000 items. A binary search checks the same sorted list. Approximately how many checks does each need in the worst case?	**Linear search needs up to 1,000,000 checks. Binary search needs only about 20 checks, because $\log_2(1{,}000{,}000) \approx 20$.** — Linear search: worst case checks every item = 1,000,000 comparisons. Binary search: halves each time, so 1,000,000 → 500,000 → 250,000 → ... → 1. That's $\log_2(1{,}000{,}000) \approx 20$ comparisons. Binary search is 50,000 times faster! But it requires the list to be sorted first.
Describe how bubble sort works. What is its time complexity?	**Bubble sort repeatedly compares adjacent elements and swaps them if they're in the wrong order, making multiple passes until no swaps are needed. Its time complexity is O(n^2).** — Bubble sort: compare pairs (1st & 2nd, 2nd & 3rd, ...), swap if out of order, repeat until sorted. Each pass moves the largest unsorted element to its correct position. It needs up to n-1 passes of up to n-1 comparisons each, giving O(n^2). It's simple but slow for large lists.

What is a stable sorting algorithm, and why does stability matter?	**A stable sort preserves the relative order of elements with equal keys. It matters when sorting by multiple criteria — for example, sorting students first by name, then by grade, where students with the same grade stay in alphabetical order.** — Stability means equal elements maintain their original relative order. Example: sorting employees by department. If Alice and Bob are both in 'Engineering' and Alice appeared first originally, a stable sort keeps Alice before Bob. Merge sort and insertion sort are stable; selection sort and quicksort (typically) are not.
What does 'greedy algorithm' mean?	**An algorithm that makes the locally optimal choice at each step** — See the correct answer above for the explanation.
Design an algorithm to find the second-largest number in a list. What is its time complexity?	**Track two variables: largest and second_largest. Scan through the list once, updating both as needed. Time complexity: O(n) since you examine each element once.** — Algorithm: Set largest = second_largest = -infinity. For each element x: if x > largest, set second_largest = largest, then largest = x. Else if x > second_largest (and x ≠ largest), set second_largest = x. This uses one pass (O(n)) and constant extra space (O(1)). Much faster than sorting (O(n log n)) for this specific task.
What is a greedy algorithm? Give an example.	**A greedy algorithm makes the locally optimal choice at each step, hoping to find a global optimum. Example: making change with the fewest coins by always picking the largest coin that fits.** — Greedy algorithms always choose the option that looks best at the moment. The coin change example: for 67¢ with US coins, greedily picking the largest coin each time (25, 25, 10, 5, 1, 1) gives 6 coins, which is optimal. However, greedy doesn't always give the best answer — some problems require exploring multiple paths.
Explain what it means for a problem to be 'NP-complete' in simple terms.	**An NP-complete problem is one where no known fast algorithm exists, but if you're given a proposed solution, you can verify it quickly. If a fast algorithm is found for any NP-complete problem, it would solve all of them.** — NP-complete problems have two properties: (1) Solutions can be verified quickly (in polynomial time). (2) They are among the hardest problems in NP — every NP problem can be transformed into an NP-complete problem. The P vs NP question (are these problems actually solvable quickly?) is the biggest open problem in computer science, with a $1 million prize.
What is the Traveling Salesman Problem (TSP), and why is it computationally difficult?	**TSP asks for the shortest route visiting every city exactly once and returning home. It's difficult because the number of possible routes grows factorially — checking all routes for 20 cities would take millions of years.** — With n cities, there are (n-1)!/2 unique routes. For 20 cities: $19!/2 \approx 6 \times 10^{16}$ routes. Even checking a billion routes per second would take about 2 years. TSP is an 'NP-hard' problem — no known algorithm solves it efficiently for all cases. Approximate solutions and heuristics are used in practice.
What is the time complexity of checking every element in a list of n items?	**O(n)** — See the correct answer above for the explanation.
What is the key insight behind dynamic programming, and how is it different from simple recursion?	**Dynamic programming solves complex problems by breaking them into overlapping subproblems, solving each only once, and storing results. Unlike simple recursion, it avoids redundant computation by building solutions bottom-up or using memoization.** — Dynamic programming (DP) recognizes when subproblems overlap. Simple recursion re-solves these overlapping subproblems wastefully. DP stores each subproblem's result in a table (bottom-up approach) or cache (top-down memoization). Example: Fibonacci — DP computes fib(1) through fib(n) once each, while naive recursion recomputes exponentially.
What is the	**Best case is the fastest an algorithm can run (most favorable input), worst case is the**

difference between best case, worst case, and average case complexity?	**slowest (least favorable input), and average case is the typical performance across all possible inputs.** — Linear search for item in a list of n elements: Best case O(1) — item is first. Worst case O(n) — item is last or absent. Average case O(n/2) = O(n) — on average, check half the list. Big O notation typically describes worst case because it provides a guarantee on maximum running time.
What does Big O notation measure?	**Big O notation describes how an algorithm's running time or space grows as the input size increases, focusing on the worst-case scenario.** — Big O notation classifies algorithms by their growth rate. O(1) is constant time (instant regardless of input). O(n) is linear (doubles with doubled input). $O(n^2)$ is quadratic (quadruples with doubled input). It focuses on worst-case behavior and ignores constants because growth rate matters most for large inputs.
Why is a binary search faster than a linear search for sorted data?	**It eliminates half the remaining elements each step** — See the correct answer above for the explanation.

9. Elements & Principles Deep Dive

✂ Question (fold →)	Answer
In the famous Königsberg Bridge Problem, people tried to walk across all 7 bridges exactly once and return home. Euler proved this was impossible. What graph property did he use?	**Euler showed that for such a walk (Eulerian circuit) to exist, every vertex must have an even degree, but the Königsberg graph had vertices with odd degrees.** — Euler's breakthrough was realizing that for a circuit visiting every edge exactly once (Eulerian circuit), every vertex must have even degree. This is because each time you pass through a vertex, you use one edge to enter and one to leave — always in pairs. Königsberg had 4 vertices, all with odd degrees (3, 3, 3, 5), making the circuit impossible. This 1736 proof is considered the birth of graph theory!
Design a graph to model the layout of your school. Identify what the vertices and edges represent, and explain one question that analyzing this graph could answer.	**Vertices could be rooms (classrooms, gym, cafeteria, office), edges could be hallways connecting them, and analyzing the graph could find the shortest path between any two rooms for evacuation planning.** — A school graph might have rooms as vertices and hallways/doors as edges. Edge weights could represent walking time. Useful questions include: What's the shortest evacuation route from each classroom to an exit? Which hallway, if blocked, would disconnect the most rooms? How could adding one new hallway most improve traffic flow? This is real facility planning using graph theory!
What is the base case in the recursive definition of n! (n factorial)?	**0! = 1** — See the correct answer above for the explanation.
What does it mean for two numbers to be coprime (relatively prime)?	**Two numbers are coprime if their greatest common divisor (GCD) is 1, meaning they share no common factors other than 1.** — Coprime (or relatively prime) numbers share no common factor except 1. Examples: 8 and 15 (factors of 8: 1,2,4,8; factors of 15: 1,3,5,15 — only 1 in common). Note: the numbers themselves don't need to be prime. Even 4 and 9 are coprime (GCD = 1).
Why is a tree (in graph theory) always a connected graph with no cycles?	**A tree connects all vertices using the minimum number of edges (n-1 edges for n vertices) without forming any loops.** — A tree with n vertices has exactly n-1 edges — the minimum needed for connectivity. Adding any edge would create a cycle; removing any edge would disconnect it. Trees are everywhere in computer science: file systems (folders within folders), HTML document structure, tournament brackets, and decision trees. They're the most efficient way to connect all points.
A pizza shop offers 3 sizes, 4 crust types, and 6 toppings. If you choose one of each, how many different pizza combinations are possible?	**There are 72 different combinations because 3 x 4 x 6 = 72.** — The Fundamental Counting Principle says to multiply the number of options at each step. 3 sizes x 4 crusts x 6 toppings = 72 total combinations.
If Set A = {1, 3, 5} and Set B = {2, 3, 4}, what is A ∩ B (the intersection)?	**{3}** — See the correct answer above for the explanation.
A library sorts books by author last name. New books arrive in random order. Which sorting strategy would be most practical: re-sorting all books from	**Inserting each new book into its correct position (insertion approach) is more practical because most books are already sorted.** — When most data is already sorted, insertion is far more efficient than re-sorting everything. This is why insertion sort performs well on nearly-sorted data. Libraries, filing systems, and databases all

scratch each time, or inserting each new book into its correct position?	use this principle — maintain sorted order incrementally rather than rebuilding from scratch.
The Handshaking Lemma says the sum of all vertex degrees in a graph equals twice the number of edges. Use this to determine if a graph can exist with 4 vertices having degrees 3, 2, 2, and 1.	**No, this graph cannot exist because the sum of degrees is 3 + 2 + 2 + 1 = 8, and 8/2 = 4 edges, but a vertex of degree 3 must connect to 3 others, and the degree-1 vertex can only connect to one — the constraints conflict.** — The degree sum is 8 (even), suggesting 4 edges could work. But let's try: vertex A (degree 3) connects to B, C, and D. Then B already has degree 1 (from A), C has degree 1, D has degree 1. We need B at degree 2, so B needs one more edge. B connects to C: B is now degree 2 ✓, C is degree 2 ✓. D is at degree 1 ✓, A is at degree 3 ✓. Actually, it DOES work! The graph A-B, A-C, A-D, B-C is valid.
Your teacher says the Fibonacci sequence (1, 1, 2, 3, 5, 8, 13, ...) appears everywhere in nature. A classmate says this is just a coincidence. Who do you think makes a stronger argument, and what evidence supports your view?	**The teacher makes a stronger argument because Fibonacci numbers appear in sunflower seed spirals, pinecone scales, flower petals, and shell curves, suggesting a deep connection to natural growth patterns.** — The teacher has strong evidence. Fibonacci numbers appear in: sunflower seed spirals (usually 34 and 55), pine cone scales (8 and 13), flower petals (lilies have 3, buttercups 5, daisies often 34 or 55), and nautilus shell growth. This isn't coincidence — these numbers emerge from efficient growth and packing algorithms in biological systems.
The Sierpinski Triangle starts with one triangle and recursively divides it. After 4 iterations, how many small triangles are shaded?	**After 4 iterations, there are 3^4 = 81 shaded triangles, because each iteration triples the number of shaded triangles.** — The Sierpinski Triangle is a fractal. At each iteration, each shaded triangle becomes 3 smaller shaded triangles. Starting with 1: after step 1 = 3, step 2 = 9, step 3 = 27, step 4 = 81. The formula is 3^n after n iterations.
What is a set in mathematics?	**A set is a well-defined collection of distinct objects or elements.** — In mathematics, a set is simply a collection of distinct objects considered as a whole. Sets can contain numbers, letters, colors, or anything else. The important rule is that each element appears only once in a set — there are no duplicates.
What is mathematical induction, and why is it useful?	**Mathematical induction is a proof technique that proves a statement is true for all natural numbers by showing it holds for a base case and that if it holds for n, it also holds for n+1.** — Induction has two steps: (1) Base case: prove the statement for n=1 (or the smallest case). (2) Inductive step: assume it's true for some n (inductive hypothesis), then prove it for n+1. Like dominoes, this chain reaction proves it for all natural numbers.
In a weighted graph of cities, the edges show distances in miles: A-B = 10, B-C = 15, A-C = 30, B-D = 20, C-D = 5. What is the shortest path from A to D?	**The shortest path from A to D is A → B → C → D with a total distance of 30 miles (10 + 15 + 5).** — Possible paths: A→B→D = 10+20 = 30, A→C→D = 30+5 = 35, A→B→C→D = 10+15+5 = 30. Both A→B→D and A→B→C→D have length 30, but A→B→C→D routes through more cities. The key insight is that the direct-looking path isn't always shortest — this is why GPS algorithms check many routes. Dijkstra's algorithm solves this efficiently.
Can a graph with 5 vertices all having odd degree exist?	**No, the sum of all degrees must be even** — See the correct answer above for the explanation.
	A conditional statement is an 'if-then' statement, written as 'if P, then Q' where P is the hypothesis and Q is the conclusion. — Conditional statements (if P then Q, written P → Q) are fundamental to logic and programming. P is called the

What is a conditional statement in logic?	hypothesis (or antecedent) and Q is the conclusion (or consequent). Surprisingly, 'if P then Q' is only false when P is true and Q is false. If P is false, the conditional is automatically true regardless of Q.
If a password must be 2 digits long and each digit is 0-9, how many possible passwords are there?	**100** — See the correct answer above for the explanation.
If P is true and Q is false, what is the truth value of P AND Q?	**False** — See the correct answer above for the explanation.
A bag contains 5 red and 3 blue marbles. You draw 2 marbles without replacement. What is the probability that you get one of each color?	**The probability is 15/28, found by calculating: (5/8)(3/7) + (3/8)(5/7) = 15/56 + 15/56 = 30/56 = 15/28.** — Two paths: (Red then Blue) = (5/8)(3/7) = 15/56, and (Blue then Red) = (3/8)(5/7) = 15/56. P(one of each) = 15/56 + 15/56 = 30/56 = 15/28 ≈ 53.6%.
Draw a graph representing the friendships in a group: Amy is friends with Ben and Cal. Ben is friends with Amy and Dana. Cal is friends with Amy only. Dana is friends with Ben only. How many vertices and edges does this graph have?	**The graph has 4 vertices (Amy, Ben, Cal, Dana) and 3 edges (Amy-Ben, Amy-Cal, Ben-Dana).** — Four people = 4 vertices. The friendships are: Amy-Ben, Amy-Cal, and Ben-Dana = 3 edges. Note that Amy is friends with Ben is the SAME edge as Ben is friends with Amy (undirected graph). Amy has degree 2 (connected to Ben and Cal), Ben has degree 2 (Amy and Dana), Cal has degree 1, and Dana has degree 1. The sum of degrees (2+2+1+1=6) equals twice the edges (2×3=6) — the Handshaking Lemma!
In a class survey, Set M = students who like math, Set S = students who like science, and Set A = students who like art. If 5 students like all three subjects, where would they appear in a three-circle Venn diagram?	**Students who like all three subjects appear in the center region where all three circles overlap.** — A three-circle Venn diagram has 7 distinct regions: 3 exclusive regions, 3 two-way overlaps, and 1 three-way overlap in the center. Students who like all three subjects belong to the intersection M ∩ S ∩ A, which is the center region. Three-circle Venn diagrams are powerful tools for analyzing complex relationships.
The Fibonacci sequence starts 1, 1, 2, 3, 5, 8, 13, ... What are the next two numbers?	**The next two numbers are 21 and 34, because each number is the sum of the previous two.** — In the Fibonacci sequence, F(n) = F(n-1) + F(n-2). So the next number is 8 + 13 = 21, and the one after is 13 + 21 = 34. The sequence continues: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...
In logic, what does the AND operation mean?	**The AND operation is true only when both statements connected by it are true.** — The AND operation (also written as ∧) requires both conditions to be true. 'It's sunny AND it's warm' is only true when BOTH conditions hold. If either one is false, the whole AND statement is false. This is used everywhere in programming: 'if (age >= 13 AND hasPermission)' requires both conditions.
What does the word 'permutation' mean in mathematics?	**A permutation is an arrangement of objects where the order matters.** — A permutation is an ordered arrangement. The sequence A-B-C is a different permutation than C-B-A, even though they use the same letters. This contrasts with combinations, where order doesn't matter. Permutations are used in passwords,

	lock codes, race finishes, and seating arrangements.
Your friend claims: 'If NOT (P AND Q) is true, then both P and Q must be false.' Build a truth table to evaluate this claim.	**The claim is false. NOT (P AND Q) is true whenever at least one is false — it includes (T,F), (F,T), and (F,F), not just (F,F).** — Truth table: P AND Q is true only for (T,T). NOT (P AND Q) flips this: false for (T,T), true for (T,F), (F,T), and (F,F). So the claim is wrong — NOT (P AND Q) is true in THREE cases, not just when both are false. By De Morgan's Law, NOT (P AND Q) = (NOT P) OR (NOT Q), which is a much weaker condition than requiring both to be false.

10. Cultural & Historical Context

✂ Question (fold →)	Answer
Your friend says bubble sort is the best sorting algorithm because it's the easiest to understand. Do you agree? What factors besides simplicity should we consider when choosing a sorting algorithm?	**While bubble sort is easy to understand, we should also consider speed, the size of the data, and whether the data is already partially sorted.** — Choosing an algorithm involves trade-offs: speed (time complexity), memory usage (space complexity), simplicity, and the nature of the data. Bubble sort is $O(n^2)$ and slow for large datasets but fine for small ones. Merge sort is $O(n \log n)$ and much faster for large datasets but uses more memory. The 'best' algorithm depends on the situation.
A network has 5 computers connected by cables. One cable fails, disconnecting the network. An engineer says adding just one more cable would prevent any single cable failure from disconnecting the network. Is this always possible?	**Not always. If the original network is a tree (minimum connections), one extra cable creates one cycle, but there might still be bridges whose failure disconnects the network.** — It depends on the topology. A 5-vertex tree has 4 edges; adding a 5th creates exactly one cycle. Vertices in that cycle gain a backup path, but vertices outside it don't. To make every edge failure survivable, the graph needs to be '2-edge-connected,' meaning every pair of vertices has at least 2 edge-disjoint paths. This requires careful analysis, not just adding random cables.
Why is the statement 'If it's Tuesday, then I have math class' different from 'If I have math class, then it's Tuesday'?	**They're different because the second is the converse of the first, and a statement and its converse can have different truth values.** — The original says Tuesday guarantees math class. The converse says math class guarantees it's Tuesday — but you might have math on other days too! A statement and its converse are NOT logically equivalent. This is a common reasoning error. The contrapositive ('If I don't have math, then it's not Tuesday') IS equivalent to the original.
How many ways can you arrange 4 different books on a shelf?	**There are 24 ways to arrange 4 books, since 4! = 4 x 3 x 2 x 1 = 24.** — Each position reduces the number of available books by one. The first slot has 4 choices, the second has 3, the third has 2, and the last has 1. Multiplying gives 4! = 24.
What is a one-way function, and why is it important for cryptography?	**A one-way function is easy to compute in one direction but extremely difficult to reverse. It's the basis of public-key cryptography — anyone can encrypt, but only the key holder can decrypt.** — One-way functions are computational trapdoors: f(x) is quick to compute, but finding x from f(x) is impractical. Prime multiplication is one-way: 317 x 331 = 104,927 is instant, but factoring 104,927 back into 317 x 331 takes much longer. This asymmetry enables RSA and other public-key systems.
What comes next in the pattern: 2, 6, 18, 54, ...?	**162** — See the correct answer above for the explanation.
A delivery company needs to visit 6 stores and return to the warehouse. Why is finding the shortest route through all stores so much harder than finding the shortest route between just two stores?	**With 6 stores, there are 720 possible orderings (6! = 720) to check, and the number grows factorially with more stops, making it far harder than a two-point shortest path.** — This is the Traveling Salesman Problem (TSP), one of the most famous problems in computer science. For 6 stores: 6! = 720 routes. For 20 stores: 20! ≈ 2.4 quintillion routes. Even the fastest computers can't check them all for large numbers. Two-point shortest path (like GPS) is solved efficiently by Dijkstra's algorithm, but TSP has no known fast solution — it's 'NP-hard.' Real delivery companies use approximation algorithms that find good (but not necessarily perfect) routes.

What pattern do you see in this sequence: 1, 1, 2, 3, 5, 8, 13, ...? What are the next two numbers?	**Each number is the sum of the two before it (the Fibonacci sequence), so the next two numbers are 21 and 34.** — This is the Fibonacci sequence, one of the most famous patterns in mathematics. Each term is the sum of the two preceding terms: 8 + 13 = 21, and 13 + 21 = 34. Fibonacci numbers appear in nature — sunflower spirals, pine cone patterns, and shell curves all follow this sequence.
If Set A = {1, 2, 3, 4, 5} and Set B = {3, 4, 5, 6, 7}, what is the intersection of A and B?	**The intersection of A and B is {3, 4, 5}.** — The intersection of two sets (written A ∩ B) contains only the elements that appear in BOTH sets. Numbers 3, 4, and 5 are in both Set A and Set B. The intersection is like the overlapping region of a Venn diagram.
What is a truth table?	**A truth table is a chart that shows every possible combination of true/false inputs and the resulting output for a logical expression.** — A truth table systematically lists every possible combination of input values and shows what the output is for each. With 2 inputs, there are 4 rows (TT, TF, FT, FF). With 3 inputs, there are 8 rows. Truth tables are used to verify logical arguments, design digital circuits, and debug computer programs.
A recursive sequence is defined as: a(1) = 1, a(2) = 1, a(n) = a(n-1) + 2 x a(n-2). Find a(5).	**a(5) = 11. Working out: a(3) = 1 + 2(1) = 3, a(4) = 3 + 2(1) = 5, a(5) = 5 + 2(3) = 11.** — a(1) = 1, a(2) = 1. a(3) = a(2) + 2 x a(1) = 1 + 2(1) = 3. a(4) = a(3) + 2 x a(2) = 3 + 2(1) = 5. a(5) = a(4) + 2 x a(3) = 5 + 2(3) = 5 + 6 = 11.
What is the difference between a recursive formula and an explicit (closed-form) formula for the sequence 2, 4, 8, 16, 32, ...?	**The recursive formula is a(1) = 2, a(n) = 2 x a(n-1). The explicit formula is a(n) = 2^n. Both define the same sequence differently.** — The recursive formula a(n) = 2 x a(n-1) tells you each term from the previous one. The explicit formula a(n) = 2^n lets you compute any term directly. For the 10th term: recursive requires computing terms 1-9 first, but explicit gives 2^10 = 1024 immediately.
Use the domino analogy to explain why both parts of mathematical induction are necessary.	**The base case is like pushing the first domino. The inductive step ensures each domino knocks down the next. Without either part, the chain fails.** — Without the base case (pushing the first domino), nothing starts falling even if dominoes are perfectly arranged. Without the inductive step (dominoes correctly spaced), pushing the first domino might only knock down a few. You need both: a starting point AND a guarantee that each case triggers the next.
Why can a graph with 5 vertices and 11 edges not exist if the graph is simple (no self-loops or parallel edges)?	**A simple graph with 5 vertices can have at most 10 edges, calculated as 5 × 4 / 2 = 10, so 11 edges is impossible.** — In a simple graph (no duplicate edges, no self-loops), each pair of vertices can have at most one edge. With 5 vertices, the number of possible pairs is C(5,2) = 5×4/2 = 10. This maximum is called a complete graph (K_5). Since 11 > 10, such a graph cannot exist. This upper bound is fundamental in graph theory.
How many 3-letter 'words' (real or nonsense) can you make from the letters A, B, C, D, E if no letter can be repeated?	**You can make 60 three-letter words, calculated as 5 × 4 × 3 = 60.** — Without repetition: 5 choices for the first letter, 4 for the second (one is used), and 3 for the third. So 5 × 4 × 3 = 60 arrangements. This is written as P(5,3) or 5!/(5-3)! = 120/2 = 60. If repetition were allowed, it would be 5 × 5 × 5 = 125.
How do we write the set of all even numbers between 1 and 11 using set notation with curly braces?	**The set of even numbers between 1 and 11 is written as {2, 4, 6, 8, 10}.** — Sets are written using curly braces { } with elements separated by commas. The even numbers between 1 and 11 are 2, 4, 6, 8, and 10. Notice we don't include 1 or 11 (they're odd), and we don't go past 10 since 12 is beyond 11.
You draw two cards from a	

standard 52-card deck without replacement. What is the probability that both are hearts?	**The probability is (13/52) x (12/51) = 156/2652 = 1/17.** — The first card has a 13/52 chance of being a heart. If it is, there are now 12 hearts left in 51 remaining cards. P(both hearts) = (13/52) x (12/51) = 156/2652 = 1/17.
What is the negation of the statement 'All birds can fly'?	**Some birds cannot fly** — See the correct answer above for the explanation.
What is a weighted graph, and when would you use one?	**A weighted graph assigns a numerical value to each edge, representing cost, distance, time, or another quantity.** — In a weighted graph, each edge has a numerical weight. Road maps use distance weights for navigation. Airline routes use time or cost weights for booking. Network diagrams use bandwidth weights for routing data. Finding the shortest (lowest total weight) path in a weighted graph is one of the most practical problems in computer science — GPS devices solve this millions of times daily.
In a video game, a treasure chest opens when the player 'hasKey AND (level >= 5 OR isPremium).' If a player has a key and is level 3 but is a premium member, does the chest open?	**Yes, the chest opens because hasKey is true, and isPremium is true which makes the OR clause true, so the whole AND expression is true.** — Work from the inside out. First: (level >= 5 OR isPremium) = (false OR true) = true. Then: hasKey AND true = true AND true = true. The chest opens! The OR inside the parentheses means either condition suffices. This is exactly how game programmers and app developers write conditional logic.
What is the sum of the first n natural numbers? (1 + 2 + 3 + ... + n)	**The sum is n(n+1)/2. For example, 1 + 2 + 3 + ... + 10 = 10(11)/2 = 55.** — Young Gauss discovered this by pairing: 1+n, 2+(n-1), 3+(n-2), etc. Each pair sums to (n+1), and there are n/2 pairs. Total = n(n+1)/2. This is also the nth triangular number. For n=100: 100(101)/2 = 5050.
Two students debate: Student A says 'Graphs are just dots and lines — they can't represent anything complex.' Student B says 'Graphs can model almost any relationship.' Who makes the stronger argument?	**Student B is correct. Despite their simplicity, graphs model social networks, transportation systems, the internet, biological pathways, language relationships, and many more complex systems.** — Student B is right. The simplicity of graphs is their power — by abstracting complex systems into vertices and edges, we can apply the same mathematical tools to social networks, airline routes, disease spread, food webs, circuit design, and the internet. Google's PageRank, GPS navigation, social media recommendations, and epidemiological models all use graph algorithms.
What is a path in graph theory?	**A path is a sequence of vertices connected by edges where no vertex is visited more than once.** — A path visits vertices in sequence without repeating any vertex. The length of a path is the number of edges traversed. If a path starts and ends at the same vertex, it's called a cycle. Finding paths between vertices is crucial for GPS navigation, network routing, and solving mazes.
If you add any two even numbers, the result is always even. Use divisibility to explain why.	**An even number can be written as 2a and another as 2b. Their sum is 2a + 2b = 2(a + b), which is 2 times an integer, so it's always even.** — Any even number can be written as 2k for some integer k. If we have 2a and 2b, then 2a + 2b = 2(a+b). Since (a+b) is an integer, 2(a+b) is divisible by 2, making it even. This is a direct proof using the definition of even numbers.
In graph theory, what is the degree of a vertex?	**The number of edges connected to it** — See the correct answer above for the explanation.

11. Creative Process & Critique

✂ Question (fold →)	Answer
Create a recursive rule for a sequence where each term is the average of the two preceding terms, starting with a(1) = 0 and a(2) = 10. Find the first 6 terms.	**Rule: a(n) = (a(n-1) + a(n-2)) / 2. Terms: 0, 10, 5, 7.5, 6.25, 6.875. The sequence converges toward 6.667 (which is 20/3).** — a(1)=0, a(2)=10, a(3)=(10+0)/2=5, a(4)=(5+10)/2=7.5, a(5)=(7.5+5)/2=6.25, a(6)=(6.25+7.5)/2=6.875. The sequence oscillates and converges to the weighted average 2/3 x 10 = 6.667... This shows how recursive averaging creates convergent behavior.
Invent a new sorting game for your classmates. Describe what objects you'd sort, what property you'd sort by, and which sorting algorithm (bubble, selection, or insertion) your game would use. Explain why you chose that algorithm.	**A good game might sort playing cards by number using insertion sort, since players naturally pick up cards one at a time and place them in order, making it intuitive and fun.** — Designing a sorting game requires creativity and understanding of algorithms. Insertion sort works well for card games because it mirrors how people naturally organize. Selection sort works for relay races where teams send someone to find the 'minimum' item. Bubble sort works for partner activities where neighbors compare. The best game choice depends on what's fun AND educational.
Social media platforms show you 'People You May Know' suggestions. How might a graph algorithm find these suggestions by looking at friend-of-friend connections?	**The algorithm looks for people who share many mutual friends with you (vertices two edges away) but aren't directly connected to you yet.** — In the social graph, your direct friends are at distance 1. People at distance 2 (friends of friends) are candidates for 'People You May Know.' The algorithm counts how many mutual friends you share — more shared neighbors = stronger recommendation. This is essentially counting common neighbors in graph theory, one of the simplest but most powerful graph algorithms used in practice.
What is 23 mod 7? What is 100 mod 12?	**23 mod 7 = 2 (because 23 = 3 x 7 + 2), and 100 mod 12 = 4 (because 100 = 8 x 12 + 4).** — 23 ÷ 7 = 3 with remainder 2, so 23 mod 7 = 2. 100 ÷ 12 = 8 with remainder 4, so 100 mod 12 = 4. The mod operation always gives a result between 0 and (divisor - 1).
In a card game, you pick up cards one at a time and slide each into its correct position among the cards already in your hand. Which sorting algorithm does this resemble?	**This resembles insertion sort, where each new element is inserted into its correct position in the sorted portion.** — Insertion sort works exactly like sorting cards in your hand. You take one card at a time and slide it into the correct position among the cards you've already sorted. It's efficient for small or nearly-sorted datasets, and it's the most intuitive sorting algorithm because people naturally use it.
What is Euler's totient function ϕ(n), and what is ϕ(12)?	**ϕ(n) counts the numbers from 1 to n that are coprime to n (share no common factors other than 1). ϕ(12) = 4, because 1, 5, 7, and 11 are coprime to 12.** — ϕ(12) counts integers from 1 to 12 that share no common factor with 12 (GCD = 1). Numbers 2,3,4,6,8,9,10,12 share a factor with 12. The coprime numbers are 1, 5, 7, 11 → ϕ(12) = 4. Euler's totient is central to RSA encryption's key generation.
How many 3-letter codes can be made from the letters A, B, C, D, E if no letter can be repeated?	**There are 60 codes because 5 x 4 x 3 = 60.** — Since order matters (ABC is different from BAC) and no repetition is allowed, this is P(5,3) = 5 x 4 x 3 = 60. The first position has 5 choices, the second has 4, and the third has 3.
A graph has vertices A, B, C, D,	**The shortest path from A to C is A → B → C, which has length 2 (two edges).** — The graph forms a cycle A-B-C-D-E-A (a pentagon). From A to C, the clockwise path

E with edges A-B, B-C, C-D, D-E, and E-A. What is the shortest path from A to C?	A→B→C has 2 edges, while the counterclockwise path A→E→D→C has 3 edges. The shortest path has length 2. There is no direct A-C edge, so length 1 is impossible. Finding shortest paths is one of the most important problems in graph theory.
Design a carnival game that is fair (expected value of $0 for the player). The game uses a spinner with 4 equal sections.	**One fair design: charge $1 to play, one section wins $4, three sections win $0. EV = (1/4)($4) - $1 = $1 - $1 = $0.** — For a fair game, EV must equal $0. With 4 equal sections and a $1 entry fee: if 1 section wins $4 and 3 win $0, then EV = (1/4)($4) + (3/4)($0) - $1 = $1 - $1 = $0. Players break even on average.
In a truth table, how many rows are needed to evaluate a compound statement with 3 variables?	**8** — See the correct answer above for the explanation.
What is a cycle in a graph?	**A path that starts and ends at the same vertex without repeating edges** — See the correct answer above for the explanation.
Look at this Venn diagram: Circle A has {1, 3, 5}, the overlap has {7, 9}, and Circle B has {2, 4, 6}. What can you conclude about the odd numbers in this diagram?	**All odd numbers (1, 3, 5, 7, 9) are in Set A, while Set B contains only even numbers.** — Every odd number in the diagram (1, 3, 5, 7, 9) is inside Circle A — some exclusively in A, and some shared with B. Circle B's exclusive elements (2, 4, 6) are all even. This pattern suggests Set A might be defined as odd numbers while Set B is defined as even numbers, with 7 and 9 fitting both criteria.
Create a counting problem about planning a class trip. Include at least three independent choices that must be made, and calculate the total number of possible trip plans.	**A good problem might be: Choose a destination (4 options), a bus type (3 options), and a lunch plan (5 options), giving 4 × 3 × 5 = 60 different trip plans.** — The multiplication principle extends to any number of independent choices. A trip plan might involve: destination (museum, park, zoo, factory = 4), transportation (school bus, coach bus, van = 3), and lunch (pizza, sandwiches, wraps, salad bar, hot dogs = 5). Total: 4 × 3 × 5 = 60 unique trip plans. Creating your own counting problems deepens understanding of the underlying structure.
Compare the truth tables for P OR Q and P AND Q. In how many of the 4 input combinations do they give different results?	**They give different results in 2 of the 4 combinations: when one input is true and the other is false.** — AND and OR agree on (T,T)→both true, and (F,F)→both false. They differ on (T,F): OR=true, AND=false, and (F,T): OR=true, AND=false. So they differ in exactly 2 of 4 cases. This shows AND is stricter than OR — AND requires unanimity while OR only needs one vote.
Find the pattern and fill in the blank: 1, 4, 9, 16, 25, __, 49.	**The missing number is 36, because these are perfect squares: 1^2, 2^2, 3^2, 4^2, 5^2, 6^2, 7^2.** — These are perfect squares: $1^2 = 1$, $2^2 = 4$, $3^2 = 9$, $4^2 = 16$, $5^2 = 25$, $6^2 = 36$, $7^2 = 49$. The missing value is $6^2 = 36$. Notice the differences between consecutive squares form their own pattern: 3, 5, 7, 9, 11, 13 — always increasing by 2. This is because $(n+1)^2 - n^2 = 2n + 1$.
What is the difference between a directed graph and an undirected graph?	**In a directed graph, edges have a direction (like one-way streets), while in an undirected graph, edges go both ways.** — In an undirected graph, friendship is mutual: if A is connected to B, then B is connected to A. In a directed graph (digraph), connections can be one-way: you can follow a celebrity on social media without them following you back. Road networks use directed graphs for one-way streets and undirected for two-way roads.
A weather app says: 'If temperature > 90°F, then display heat warning.' The contrapositive is: 'If no heat	**Yes, a conditional statement and its contrapositive are logically equivalent — they always have the same truth value.** — A statement and its contrapositive

warning is displayed, then temperature ≤ 90°F.' Are both statements always true at the same time?	are ALWAYS logically equivalent. If 'A → B' is true, then 'NOT B → NOT A' must also be true. This is different from the converse ('if heat warning, then temp > 90') which might be false — maybe the system has a bug. The contrapositive is a powerful reasoning tool in mathematics, programming, and everyday logic.
Build a truth table for NOT (P OR Q). How many rows does it have, and when is the result true?	**The truth table has 4 rows (for TT, TF, FT, FF), and NOT (P OR Q) is true only when both P and Q are false.** — With 2 variables, the truth table has 2^2 = 4 rows. P OR Q is true for (T,T), (T,F), (F,T) and false only for (F,F). Applying NOT flips everything: NOT(P OR Q) is false for the first three and true only for (F,F). This equals (NOT P) AND (NOT Q) — that's De Morgan's Law in action!
Why is a binary search faster than a linear search for sorted data?	**It eliminates half the remaining elements each step** — See the correct answer above for the explanation.
What are the Lucas numbers, and how are they similar to and different from the Fibonacci numbers?	**Lucas numbers follow the same recursive rule as Fibonacci (add the previous two) but start with different base cases: L(1) = 2, L(2) = 1, giving 2, 1, 3, 4, 7, 11, 18, ...** — Lucas numbers: 2, 1, 3, 4, 7, 11, 18, 29, 47, ... Same rule as Fibonacci (L(n) = L(n-1) + L(n-2)) but different starting values (2 and 1 instead of 1 and 1). This shows how base cases dramatically change a recursive sequence's values while maintaining the same pattern.
How many different ways can you arrange the letters A, B, and C in a row?	**6** — See the correct answer above for the explanation.
A cell divides into 2 cells every hour. Starting with 1 cell, write a recursive formula and find how many cells exist after 8 hours.	**Recursive formula: C(0) = 1, C(n) = 2 x C(n-1). After 8 hours: C(8) = 2^8 = 256 cells.** — Base case: C(0) = 1 cell at the start. Recursive case: C(n) = 2 x C(n-1) because each cell divides into 2. C(1)=2, C(2)=4, C(3)=8, C(4)=16, C(5)=32, C(6)=64, C(7)=128, C(8)=256.
What is the empty set, and how is it written?	**The empty set is a set with no elements, written as { } or ∅.** — The empty set (also called the null set) contains no elements at all. It can be written as { } or with the special symbol ∅. Note that {0} is NOT empty — it contains the number zero. The empty set is a subset of every other set, which makes it special in mathematics.
Design a set of rules for a simple robot using AND, OR, and NOT. The robot should: move forward if the path is clear AND battery is charged, turn left if there's an obstacle AND the left path is clear, and stop if the battery is NOT charged.	**The rules could be: moveForward = clearAhead AND charged; turnLeft = obstacleAhead AND clearLeft; stop = NOT charged. These three rules cover the basic scenarios using logical operations.** — These rules demonstrate how logical operations control real systems. The AND operations ensure multiple conditions are checked simultaneously. The NOT operation creates a safety override. In real robotics, these Boolean expressions form the basis of decision-making. Priority matters too — the stop rule (NOT charged) should override the others for safety.
What does it mean for a graph to be 'connected'?	**A connected graph has a path between every pair of vertices — you can reach any vertex from any other vertex.** — A connected graph ensures reachability — from any vertex, there exists a path to every other vertex. If a graph is disconnected, it breaks into separate 'components' with no edges between them. The internet aims to be a connected graph so any computer can reach any other. Road networks in cities are typically connected.

12. Professional Practice & Careers

✂ Question (fold →)	Answer
A school password must be exactly 3 characters: the first must be a letter (A-Z), the second must be a digit (0-9), and the third must be a special symbol chosen from {!, @, #, $, %}. A security expert says this is less secure than a 3-character password using any of these 41 characters in any position. Is the expert correct?	**Yes, the expert is correct. The restricted format gives 26 × 10 × 5 = 1,300 passwords, while the unrestricted format gives 41 × 41 × 41 = 68,921 passwords.** — The restricted format has only 26 × 10 × 5 = 1,300 options because each position is limited. The unrestricted format has 41^3 = 68,921 options — about 53 times more secure. While the restricted format seems 'stronger' because it forces different character types, it actually tells attackers exactly what to expect in each position, dramatically reducing the search space.
What is the degree of a vertex in a graph?	**The degree of a vertex is the number of edges connected to it.** — The degree of a vertex counts its connections. In a social network, a person's degree is their number of friends. A vertex with degree 0 (no connections) is called isolated. The Handshaking Lemma states that the sum of all vertex degrees equals twice the number of edges — because each edge contributes to the degree of two vertices.
What is the Chinese Remainder Theorem, explained simply?	**The Chinese Remainder Theorem says if you know the remainders of a number when divided by several coprime divisors, you can uniquely determine the remainder when divided by their product.** — If you know n mod 3 = 2 and n mod 5 = 3, then n mod 15 = 8 (since 8 mod 3 = 2 and 8 mod 5 = 3). The CRT guarantees a unique solution mod (3 x 5) = 15 when the divisors are coprime. This 1,500+ year-old theorem is used in RSA computation and error-correcting codes.
What is the complement of an event in probability?	**The complement of an event is everything that is NOT that event, and their probabilities add up to 1.** — The complement of event A (written A') includes all outcomes NOT in A. If P(A) = 0.3, then P(A') = 1 - 0.3 = 0.7. The probability of an event and its complement always sum to 1.
A student claims that if a graph has 6 vertices and each vertex has degree 2, the graph must be a single hexagon (6-cycle). Is this true?	**No, the graph could also be two separate triangles (3-cycles), since each vertex would still have degree 2.** — When every vertex has degree 2, the graph must consist of disjoint cycles (each vertex has one edge in and one edge out). With 6 vertices, the possibilities are: one 6-cycle (hexagon), one 3-cycle + one 3-cycle (two triangles), or one 4-cycle + isolated vertices with degree 2 (impossible without adjusting). The two valid options are the hexagon or two triangles. This shows that degree sequences don't uniquely determine a graph!
What is the closed-form (explicit) formula for the arithmetic sequence 3, 7, 11, 15, 19, ...?	**The explicit formula is a(n) = 4n - 1, where n starts at 1.** — The first term a(1) = 3 and the common difference d = 4. Using the formula a(n) = a(1) + (n-1)d = 3 + (n-1)(4) = 3 + 4n - 4 = 4n - 1. Verify: a(1) = 4(1) - 1 = 3, a(2) = 4(2) - 1 = 7, a(3) = 4(3) - 1 = 11.
Why does the set {3, 1, 2} equal the set {1, 2, 3}?	**The two sets are equal because sets only care about which elements they contain, not the order.** — Sets are unordered collections, meaning {3, 1, 2} and {1, 2, 3} contain the same elements and are therefore equal. This is different from sequences or lists where order matters. Two sets are equal if and only if they contain exactly the same elements.

What does it mean for two numbers to be 'coprime'?	**Their greatest common divisor is 1** — See the correct answer above for the explanation.
Why does a 3-digit lock with digits 0-9 have 1,000 possible codes instead of 30?	**Each of the 3 positions can independently be any of 10 digits, so there are 10 × 10 × 10 = 1,000 codes.** — Since digits can repeat, each of the 3 positions has 10 independent choices (0-9). By the multiplication principle: 10 × 10 × 10 = 1,000 possible codes (000 through 999). If repetition were NOT allowed, there would be only 10 × 9 × 8 = 720 codes. Whether repetition is allowed dramatically changes the count.
Look at this pattern of triangular numbers: 1, 3, 6, 10, 15, 21, 28. What is the rule for finding the next triangular number, and why are they called 'triangular'?	**Each triangular number adds one more than the previous gap (add 2, add 3, add 4, etc.), and they're called triangular because they represent dots arranged in equilateral triangles.** — Triangular numbers follow the pattern: T(n) = 1 + 2 + 3 + ... + n. The differences between consecutive terms increase by 1 each time (2, 3, 4, 5, 6, 7). They're called triangular because you can arrange that many dots into a perfect equilateral triangle: row 1 has 1 dot, row 2 has 2, row 3 has 3, and so on. The formula is T(n) = n(n+1)/2.
De Morgan's Law says NOT (P AND Q) = (NOT P) OR (NOT Q). Can you explain why this makes intuitive sense using a real-world example?	**If it's NOT the case that 'you passed the test AND did the homework,' that means you either didn't pass the test OR didn't do the homework (or both).** — De Morgan's Law captures everyday reasoning perfectly. If someone says 'You didn't BOTH clean your room and do your homework,' it means at least one wasn't done — you didn't clean OR you didn't do homework. Similarly, NOT (P OR Q) = (NOT P) AND (NOT Q): 'You didn't do either' means you didn't do the first AND you didn't do the second.
Complete the truth table for P AND Q. When P = True and Q = False, what is the result?	**When P is true and Q is false, P AND Q is false because AND requires both inputs to be true.** — AND produces true ONLY when both inputs are true. The complete truth table for P AND Q is: (T,T)→T, (T,F)→F, (F,T)→F, (F,F)→F. Only one of the four combinations gives true. This is why AND is considered a 'strict' operation — it's very hard to make an AND statement true.
What does the NOT operation do to a logical statement?	**The NOT operation reverses the truth value — it turns true into false and false into true.** — NOT (also written as ¬ or !) is the simplest logical operation — it just flips the truth value. If P is true, then NOT P is false, and vice versa. In programming, this is the '!' operator: if (isLoggedIn) checks for true, while if (!isLoggedIn) checks for false. NOT is essential for expressing negation in logic circuits, code, and mathematical proofs.
Create a logic puzzle for your classmates. Write three if-then clues that, when combined, allow the solver to determine which student (Amy, Ben, or Cal) plays which instrument (drums, flute, or guitar).	**A good puzzle might be: (1) If Amy plays drums, then Ben plays guitar. (2) Cal does NOT play flute. (3) If Ben does NOT play drums, then Amy plays flute. From clue 2, Cal plays drums or guitar. Testing possibilities leads to a unique solution.** — Logic puzzles use conditional reasoning and elimination. From clue 2: Cal plays drums or guitar. Try Cal = drums: then from clue 1's contrapositive, we explore possibilities. These puzzles develop systematic reasoning skills. The key is that each clue eliminates possibilities until only one consistent assignment remains — this is the essence of logical deduction.
What happens if a recursive definition has no base case?	**Without a base case, the recursion never stops — it creates an infinite loop that keeps calling itself forever.** — A base case is essential for recursion to terminate. Without it, the function keeps calling itself with no way to stop: f(5) calls f(4), which calls f(3), which calls f(2)... forever. In computing, this crashes the program with a 'stack overflow' error.

How many different ways can you distribute 3 identical stickers among 4 friends (each friend can get 0 or more stickers)?	**There are 20 ways, using the stars and bars method: C(3+4-1, 4-1) = C(6,3) = 20.** — Distributing n identical items among k groups uses the 'stars and bars' formula: C(n+k-1, k-1). Here, C(3+4-1, 4-1) = C(6,3) = 20. The distributions include (3,0,0,0), (2,1,0,0), (1,1,1,0), and all their rearrangements.
You flip a fair coin 3 times. What is the probability of getting exactly 2 heads?	**The probability is 3/8 because there are 3 favorable outcomes (HHT, HTH, THH) out of 8 total outcomes.** — All outcomes: HHH, HHT, HTH, THH, HTT, THT, TTH, TTT (8 total). Outcomes with exactly 2 heads: HHT, HTH, THH (3 favorable). Probability = 3/8.
Your 4-digit bike lock allows digits 0-9 with repetition. Your friend's lock also has 4 positions but uses letters A-Z (26 options) without repetition. Whose lock has more possible combinations?	**Your friend's letter lock has more combinations: 26 × 25 × 24 × 23 = 358,800, compared to your 10 × 10 × 10 × 10 = 10,000.** — Your number lock: 10^4 = 10,000 combinations. Your friend's letter lock: 26 × 25 × 24 × 23 = 358,800 combinations. Despite not allowing repetition, the letter lock wins by a factor of 35 because the larger alphabet (26 vs 10) more than compensates. This shows that the size of the symbol set matters more than whether repetition is allowed.
Why do passwords get exponentially harder to guess as they get longer? Compare a 4-character password to an 8-character password using only lowercase letters (26 options).	**A 4-character password has 26^4 ≈ 457,000 possibilities, while an 8-character password has 26^8 ≈ 209 billion, making it about 457,000 times harder to guess.** — 26^4 = 456,976 and 26^8 = 208,827,064,576. Doubling the password length doesn't double the difficulty — it SQUARES it ($26^8 = (26^4)^2$). This exponential growth is why cybersecurity experts recommend long passwords. Each additional character multiplies the total possibilities by 26, making brute-force guessing impractical.
What is a cycle in a graph, and how is it different from a path?	**A cycle is a path that starts and ends at the same vertex, forming a closed loop.** — A cycle is a closed path — it starts at a vertex, visits other vertices without repeating, and returns to the start. A triangle (3 vertices connected in a loop) is the smallest cycle. Cycles are important in detecting deadlocks in computer systems, finding loops in circuits, and identifying redundant connections in networks.
Use a truth table to verify whether 'P AND (P OR Q)' always equals P.	**Yes, P AND (P OR Q) always equals P. When P is true, the expression is true; when P is false, the expression is false — matching P exactly.** — Row by row: (T,T): T AND (T OR T) = T AND T = T = P✓. (T,F): T AND (T OR F) = T AND T = T = P✓. (F,T): F AND (F OR T) = F AND T = F = P✓. (F,F): F AND (F OR F) = F AND F = F = P✓. The result matches P in every row, proving the absorption law: P AND (P OR Q) ≡ P.
What does it mean for Set A to be a subset of Set B?	**Set A is a subset of Set B when every element in A is also found in B.** — A subset is a set entirely contained within another set. If A = {1, 2} and B = {1, 2, 3, 4}, then A is a subset of B (written A ⊆ B) because both 1 and 2 are in B. Every set is a subset of itself, and the empty set is a subset of every set.
Sort the list [5, 2, 8, 1, 9] using selection sort. What does the list look like after the first pass?	**After the first pass, the list is [1, 2, 8, 5, 9] because 1 (the smallest) swaps with 5 (the first element).** — Selection sort works by finding the smallest element in the unsorted portion and swapping it into the next sorted position. The smallest element in [5, 2, 8, 1, 9] is 1, which is at position 4. We swap it with position 1 (value 5), giving us [1, 2, 8, 5, 9]. Now position 1 is sorted.
A committee of 3 is chosen from 8 people. How many possible committees are there?	**There are 56 possible committees because C(8,3) = 8! / (3! x 5!) = 56.** — Committee members have no distinct roles, so order doesn't matter. C(8,3) = 8! / (3! x 5!) = (8 x 7 x 6) / (3 x 2 x 1) = 336/6 = 56.

Invent a board game that teaches graph theory concepts. Describe the game board, the rules, and which graph theory concept players would learn while playing.	**A good game might have players build a network of cities (vertices) and roads (edges), earning points for connecting all cities with minimum roads (spanning tree) while opponents can 'block' edges, teaching connectivity and tree concepts.** — A great graph theory game combines strategy with learning. Players could race to build spanning trees (connect all cities cheaply), find shortest paths (deliver goods efficiently), or create Eulerian circuits (visit all roads). Real games like Ticket to Ride, Pandemic, and Settlers of Catan all use graph theory — routes, connectivity, and network building are inherently fun because they mirror real decision-making.

13. Introduction to Sets

✂ Question (fold →)	Answer
Create a word problem that requires using both permutations and combinations to solve.	**Example: A teacher picks a 3-person team from 8 students (C(8,3) = 56), then assigns captain, co-captain, and recorder roles (3! = 6). Total arrangements = 56 x 6 = 336.** — Multi-step problems often combine both. First, choosing the team is a combination (order doesn't matter: C(8,3) = 56 ways). Then assigning distinct roles is a permutation (order matters: 3! = 6 ways). By the multiplication principle, total = 56 x 6 = 336.
What is 0! (zero factorial)?	**Zero factorial equals 1 by mathematical convention.** — By convention, 0! = 1. While this seems counterintuitive, it makes mathematical sense: there is exactly one way to arrange zero objects (the empty arrangement). This convention also ensures that formulas like the binomial coefficient n!/(k!(n-k)!) work correctly when k = 0 or k = n.
What is the symbol ∈ used for in set notation?	**The symbol ∈ means 'is an element of' or 'belongs to' a set.** — The symbol ∈ is read as 'is an element of' or 'belongs to.' When we write 3 ∈ {1, 2, 3, 4}, we mean that 3 is a member of that set. Conversely, ∉ means 'is not an element of,' so 5 ∉ {1, 2, 3, 4}.
You need to sort 5 cards: [8, 3, 1, 5, 2]. Show the steps using selection sort.	**Selection sort finds the minimum and places it at the front. Steps: [1,3,8,5,2] → [1,2,8,5,3] → [1,2,3,5,8] → [1,2,3,5,8] → done.** — Selection sort: Pass 1: min of [8,3,1,5,2] is 1 at index 2, swap with index 0 → [1,3,8,5,2]. Pass 2: min of [3,8,5,2] is 2 at index 4, swap with index 1 → [1,2,8,5,3]. Pass 3: min of [8,5,3] is 3 at index 4, swap with index 2 → [1,2,3,5,8]. Pass 4: min of [5,8] is 5, already in place. Done!
What is the difference between best case, worst case, and average case complexity?	**Best case is the fastest an algorithm can run (most favorable input), worst case is the slowest (least favorable input), and average case is the typical performance across all possible inputs.** — Linear search for item in a list of n elements: Best case O(1) — item is first. Worst case O(n) — item is last or absent. Average case O(n/2) = O(n) — on average, check half the list. Big O notation typically describes worst case because it provides a guarantee on maximum running time.
Write a recursive rule for the sequence: 2, 6, 18, 54, 162, ...	**The recursive rule is a(1) = 2 and a(n) = 3 x a(n-1), because each term is 3 times the previous term.** — Each term is 3 times the previous term (6/2=3, 18/6=3, 54/18=3). The recursive definition is: a(1) = 2, a(n) = 3 x a(n-1) for n > 1. This is a geometric sequence with first term 2 and common ratio 3.
Design a recursive sequence inspired by a real-world scenario (like population growth, compound interest, or a game score). Define the base case and recursive rule.	**Example — Savings account: S(0) = $100 (initial deposit), S(n) = 1.05 x S(n-1) + 10 (5% interest plus $10 monthly deposit). This models real compound growth with regular contributions.** — Many real-world processes are naturally recursive. The savings example: S(0)=$100, S(n)=1.05 x S(n-1) + 10 captures both interest growth (multiplicative) and regular deposits (additive). Other examples: bacteria doubling, loan payments, game levels, or viral sharing on social media.
You have 5 friends and want to invite exactly 2 to a sleepover. How many different pairs	**You can choose 10 different pairs, calculated as 5! / (2! x 3!) = 10.** — Since the order of choosing friends doesn't matter, this is a combination. C(5,2) = 5! / (2! x 3!) = (5 x 4) / (2 x 1) = 10 different pairs.

can you choose?	
What is an edge in graph theory?	**An edge is a connection between two vertices in a graph, representing a relationship or link.** — An edge connects two vertices, representing a relationship. In a road map graph, an edge is a road between two intersections. In a social network, an edge means two people are friends. Edges can be directed (one-way, like following someone on social media) or undirected (two-way, like a mutual friendship).
An algorithm takes 2 seconds for 100 items. If it's $O(n^2)$, approximately how long will it take for 1000 items?	**It will take approximately 200 seconds, because when n increases 10x with $O(n^2)$, the time increases 100x (10^2 = 100).** — With $O(n^2)$: time $\propto$ n^2. For n=100: 2 seconds = k × 100^2 = k × 10,000, so k = 0.0002. For n=1000: time = 0.0002 × 1000^2 = 0.0002 × 1,000,000 = 200 seconds. Input grew 10x, so time grew 10^2 = 100x. This is why $O(n^2)$ algorithms struggle with large inputs.
Find the greatest common divisor (GCD) of 48 and 36 using the Euclidean algorithm.	**GCD(48, 36) = 12. Steps: 48 = 1 x 36 + 12, then 36 = 3 x 12 + 0. When the remainder is 0, the last non-zero remainder (12) is the GCD.** — Euclidean algorithm: GCD(48,36) → 48 = 1(36) + 12, so GCD(36,12) → 36 = 3(12) + 0. Remainder is 0, so GCD = 12. The algorithm works because GCD(a,b) = GCD(b, a mod b), and it always terminates because remainders decrease.
What is the divide and conquer strategy in algorithm design?	**Divide and conquer breaks a problem into smaller subproblems, solves each one independently, then combines the solutions to solve the original problem.** — Divide and conquer has three steps: (1) Divide the problem into smaller subproblems. (2) Conquer (solve) each subproblem recursively. (3) Combine the solutions. Merge sort divides a list into halves, sorts each half (recursively), then merges the sorted halves. This gives O(n log n) instead of $O(n^2)$.
Rank these Big O complexities from fastest to slowest: $O(n^2)$, O(1), O(n log n), O(n), O(2^n).	**From fastest to slowest: O(1), O(n), O(n log n), $O(n^2)$, O(2^n).** — For n=100: O(1)=1, O(n)=100, O(n log n)≈664, $O(n^2)$=10,000, O(2^n)≈1.27×10^30. The gap between polynomial and exponential is dramatic: $O(n^2)$ is manageable for millions of items, but O(2^n) becomes impractical beyond about 30-40 items.
Your class has 20 students who like pizza and 15 students who like tacos. 8 students like both. Use set operations to find how many students like pizza OR tacos.	**There are 27 students who like pizza or tacos, calculated as 20 + 15 - 8 = 27.** — This uses the inclusion-exclusion principle. If we add 20 + 15 = 35, we've counted the 8 students who like both pizza AND tacos twice. So we subtract 8 to avoid double-counting: 20 + 15 - 8 = 27. This is one of the most useful formulas in discrete mathematics.
The Tower of Hanoi puzzle with n disks requires a minimum of 2^n - 1 moves. How many moves does a 6-disk puzzle require?	**A 6-disk Tower of Hanoi requires 63 moves, because 2^6 - 1 = 64 - 1 = 63.** — The Tower of Hanoi formula is T(n) = 2^n - 1. For 6 disks: T(6) = 2^6 - 1 = 64 - 1 = 63. This is also recursive: T(n) = 2 x T(n-1) + 1, with T(1) = 1. You move n-1 disks aside (T(n-1) moves), move the largest disk (1 move), then move n-1 disks back (T(n-1) moves).
What is the probability of rolling a sum of 7 with two	**The probability is 6/36 or 1/6, because there are 6 combinations that sum to 7 out of 36 total outcomes.** — Two dice produce 36 equally likely outcomes (6 x 6). The pairs that sum to 7 are: (1,6), (2,5), (3,4), (4,3), (5,2), (6,1) — that's 6 outcomes. So P(sum=7) = 6/36 = 1/6.

standard dice?	
Design a number sequence that follows a rule you create. Write the first 6 terms and describe the rule so a classmate could continue the pattern.	**A good sequence might be: 2, 6, 12, 20, 30, 42 where the rule is 'each term is n × (n + 1)' or equivalently 'the differences increase by 2 each time: 4, 6, 8, 10, 12.'** — Creating sequences helps you understand patterns deeply. The sequence 2, 6, 12, 20, 30, 42 uses the rule n(n+1): 1×2=2, 2×3=6, 3×4=12, 4×5=20, 5×6=30, 6×7=42. The next term would be 7×8=56. Notice these are twice the triangular numbers! Many interesting sequences hide connections to other mathematical patterns.
If Set A = {1, 3, 5, 7} and Set B = {2, 4, 6, 8}, what do we call these two sets that share no elements?	**Sets that share no elements are called disjoint sets.** — Disjoint sets are sets that have no elements in common — their intersection is empty. Set A contains only odd numbers and Set B contains only even numbers, so they cannot share any elements. This concept is important in probability and data classification.
What is the difference between a brute force approach and an optimized algorithm for finding if a number n is prime?	**Brute force checks all numbers from 2 to n-1 (O(n) checks). An optimized approach only checks up to the square root of n (O(√n) checks), because if n has a factor larger than √n, it must also have one smaller than √n.** — Brute force: test divisibility by 2, 3, 4, ..., n-1. For n=1,000,000, that's ~1 million checks. Optimized: only test up to √n ≈ 1000, since if n = a × b, one factor must be ≤ √n. Further optimization: skip even numbers after 2, reducing checks to ~500. For 1,000,000 this is 1000x faster!
What is a greedy algorithm? Give an example.	**A greedy algorithm makes the locally optimal choice at each step, hoping to find a global optimum. Example: making change with the fewest coins by always picking the largest coin that fits.** — Greedy algorithms always choose the option that looks best at the moment. The coin change example: for 67¢ with US coins, greedily picking the largest coin each time (25, 25, 10, 5, 1, 1) gives 6 coins, which is optimal. However, greedy doesn't always give the best answer — some problems require exploring multiple paths.
In logic, what does the OR operation mean?	**The OR operation is true when at least one of the statements connected by it is true.** — The OR operation (also written as ∨) is true when at least one condition is true — including when both are true. 'I'll bring an umbrella if it's raining OR cloudy' means any rain, any clouds, or both will trigger the action. Note: logical OR differs from everyday English 'or,' which often means 'one or the other but not both' (called exclusive OR).
The Collatz conjecture says: if n is even, divide by 2; if n is odd, multiply by 3 and add 1. Starting from 6, trace the sequence until it reaches 1.	**Starting from 6: 6, 3, 10, 5, 16, 8, 4, 2, 1. It takes 8 steps to reach 1.** — 6 (even→3), 3 (odd→10), 10 (even→5), 5 (odd→16), 16 (even→8), 8 (even→4), 4 (even→2), 2 (even→1). The Collatz conjecture (still unproven!) states that every positive integer eventually reaches 1 through this process.
In graph theory, what is a vertex (also called a node)?	**A vertex is a point in a graph that represents an object, person, or location.** — A vertex (plural: vertices) is a fundamental unit of a graph, representing an entity. In a friendship graph, each person is a vertex. In a road network, each intersection is a vertex. In a computer network, each device is a vertex. Vertices are connected by edges, which represent relationships between them.
A game costs $2 to play. You roll a die: if	**The expected value is -$1/3 per game (lose about 33 cents on average), so**

you roll a 6, you win \$10; otherwise you win nothing. Should you play this game?	**mathematically you should not play.** — EV = P(win) x prize - cost = (1/6)(\$10) - \$2 = \$10/6 - \$2 = \$1.67 - \$2 = -\$0.33. On average you lose about 33 cents per game. Over many plays, you'd lose money, so it's not a fair game.
What is the key insight behind dynamic programming, and how is it different from simple recursion?	**Dynamic programming solves complex problems by breaking them into overlapping subproblems, solving each only once, and storing results. Unlike simple recursion, it avoids redundant computation by building solutions bottom-up or using memoization.** — Dynamic programming (DP) recognizes when subproblems overlap. Simple recursion re-solves these overlapping subproblems wastefully. DP stores each subproblem's result in a table (bottom-up approach) or cache (top-down memoization). Example: Fibonacci — DP computes fib(1) through fib(n) once each, while naive recursion recomputes exponentially.

14. Introduction to Sets

✂ Question (fold →)	Answer
You have a sorted list of 1024 names. Using binary search, what is the maximum number of comparisons needed to find any name?	**The maximum is 10 comparisons, because $\log_2(1024) = 10$. Each comparison halves the remaining possibilities.** — Binary search halves the search space with each comparison. Starting with 1024 names: after 1 comparison = 512 remaining, after 2 = 256, ... after 10 = 1. So at most 10 comparisons. In general, binary search on n items needs at most $\lceil \log_2(n) \rceil$ comparisons. For 1024 = 2^10, that's exactly 10.
What is the time complexity of checking every element in a list of n items?	**O(n)** — See the correct answer above for the explanation.
What does 'greedy algorithm' mean?	**An algorithm that makes the locally optimal choice at each step** — See the correct answer above for the explanation.
What is the difference between symmetric and asymmetric encryption?	**Symmetric encryption uses the same key to encrypt and decrypt, while asymmetric encryption uses a public key to encrypt and a different private key to decrypt.** — Symmetric encryption (like AES) uses one shared secret key. It's fast but requires securely sharing the key. Asymmetric encryption (like RSA) uses a public key anyone can use to encrypt, but only the holder of the matching private key can decrypt. RSA's security relies on the difficulty of factoring large numbers.
A school lunch offers 4 main dishes, 3 side dishes, and 2 drinks. The school wants to know how many different meal combinations exist. If they add one more drink option, how does the total change?	**Originally there are 4 × 3 × 2 = 24 meals. Adding one drink makes it 4 × 3 × 3 = 36 meals — an increase of 12.** — Before: 4 × 3 × 2 = 24 meals. After: 4 × 3 × 3 = 36 meals. The increase is 12, which equals 4 × 3 × 1 — the number of main+side combinations times the one new drink. Adding a single option to one category multiplies through all the other categories, which is why menus with many categories grow so quickly.
In how many ways can 6 runners finish a race in first, second, and third place?	**There are 120 ways, calculated as 6 × 5 × 4 = 120.** — Any of the 6 runners could finish first. Once first is determined, 5 remain for second place. Then 4 remain for third. By the multiplication principle: 6 × 5 × 4 = 120 ways. Note that 6! = 720 would be the answer if we needed ALL 6 finishing positions, not just the top 3.
A teacher asks students to line up by height. She tells them: 'Compare yourself to the person next to you, and swap if the taller person is in front.' They repeat this until everyone is in order. What sorting algorithm is the teacher using, and why might it be slow for a class of 30?	**The teacher is using bubble sort, and it could be slow because it may need up to 435 comparisons (30 × 29 / 2) in the worst case.** — This is bubble sort — comparing adjacent elements and swapping until sorted. For 30 students, the worst case needs 30 × 29 / 2 = 435 comparisons. That's a lot of swapping! Faster algorithms like merge sort would only need about 150 comparisons (30 × $\log_2$30), which is why computer scientists study algorithm efficiency.
What are the base case and recursive case for the factorial function n!?	**The base case is 0! = 1, and the recursive case is n! = n x (n-1)! for n > 0.** — The factorial function has base case 0! = 1 (by definition). The recursive case says n! = n x (n-1)!. So 5! = 5 x 4! = 5 x 4 x 3! = 5 x 4 x 3 x 2 x 1! = 5 x 4 x 3 x 2 x 1 x 0! = 120.

Why is merge sort (O(n log n)) faster than bubble sort (O(n²)) for large lists?	**For large n, n log n grows much slower than n². With 1 million items, merge sort does about 20 million operations versus bubble sort's 1 trillion operations.** — For n = 1,000,000: $n^2 = 10^{12}$ (1 trillion), while $n \log_2 n \approx$ 20,000,000 (20 million). Merge sort is about 50,000 times faster! The O(n log n) vs O(n²) gap widens with larger inputs. For small lists (n < 50), bubble sort may actually be faster due to lower overhead.
Trace through the recursive calculation: fib(5), where fib(1) = 1, fib(2) = 1, and fib(n) = fib(n-1) + fib(n-2).	**fib(5) = fib(4) + fib(3) = (fib(3) + fib(2)) + (fib(2) + fib(1)) = ((fib(2) + fib(1)) + 1) + (1 + 1) = ((1 + 1) + 1) + 2 = 5.** — Tracing: fib(5) = fib(4) + fib(3). Then fib(4) = fib(3) + fib(2) = fib(3) + 1. And fib(3) = fib(2) + fib(1) = 1 + 1 = 2. So fib(4) = 2 + 1 = 3. Finally, fib(5) = 3 + 2 = 5.
How does binary search work, and what is its time complexity?	**Binary search works on a sorted list by repeatedly checking the middle element and eliminating half the remaining elements. Its time complexity is O(log n).** — Binary search: (1) Check middle element. (2) If it matches, done. (3) If target is smaller, search the left half. (4) If larger, search the right half. Each step halves the search space: 1000 items → 500 → 250 → 125 → ... → 1 in about 10 steps. $\log_2(1000) \approx 10$, so O(log n).
What is the Traveling Salesman Problem (TSP), and why is it computationally difficult?	**TSP asks for the shortest route visiting every city exactly once and returning home. It's difficult because the number of possible routes grows factorially — checking all routes for 20 cities would take millions of years.** — With n cities, there are (n-1)!/2 unique routes. For 20 cities: $19!/2 \approx 6 \times 10^{16}$ routes. Even checking a billion routes per second would take about 2 years. TSP is an 'NP-hard' problem — no known algorithm solves it efficiently for all cases. Approximate solutions and heuristics are used in practice.
A student says there are 24 ways to arrange the letters in the word MATH. Another student says there are also 24 ways to arrange MEET. Who is correct, and why?	**The first student is correct (MATH has 4! = 24 arrangements), but MEET has only 12 because the two E's are identical, giving 4!/2! = 12.** — MATH has 4 distinct letters, so there are 4! = 24 unique arrangements. MEET has a repeated E, so swapping the two E's doesn't create a new arrangement. We divide by 2! to account for the identical letters: 4!/2! = 24/2 = 12. This principle extends: MISSISSIPPI has 11!/(4!4!2!) = 34,650 arrangements.
What is Pascal's Triangle, and how does it relate to combinations?	**Pascal's Triangle is a triangular array where each number is the sum of the two numbers above it, and the entries represent combination values C(n,r).** — Pascal's Triangle starts with 1 at the top. Each entry is the sum of the two entries above it. Row n (starting from 0) contains the combination values C(n,0) through C(n,n). For example, row 4 is 1,4,6,4,1 which gives C(4,0) through C(4,4).
A linear search checks 1,000,000 items. A binary search checks the same sorted list. Approximately how many checks does each need in the worst case?	**Linear search needs up to 1,000,000 checks. Binary search needs only about 20 checks, because $\log_2(1{,}000{,}000) \approx 20$.** — Linear search: worst case checks every item = 1,000,000 comparisons. Binary search: halves each time, so 1,000,000 → 500,000 → 250,000 → ... → 1. That's $\log_2(1{,}000{,}000) \approx 20$ comparisons. Binary search is 50,000 times faster! But it requires the list to be sorted first.
A school rule says: 'Students can use the computer lab if they have finished homework AND it is not lunchtime.' Express this as a logical expression and evaluate it when homework is done and it IS lunchtime.	**The expression is 'finishedHomework AND NOT lunchtime.' When homework is done (true) and it is lunchtime (true), NOT lunchtime is false, so the result is false — the student cannot use the lab.** — Step by step: finishedHomework = true, lunchtime = true, NOT lunchtime = false. Then: true AND false = false. The student cannot use the lab even though homework is done, because it's lunchtime. Both conditions of the AND must be satisfied. This is how access control systems work in buildings, websites, and apps.

What does it mean for two events to be independent in probability?	**Two events are independent when the outcome of one event does not affect the probability of the other event.** — Independent events don't influence each other. For example, flipping a coin (heads/tails) doesn't change the probability of rolling a 6 on a die. Mathematically, P(A and B) = P(A) x P(B) for independent events.
How many different 4-digit PIN codes are possible using digits 0-9 if digits can be repeated?	**There are 10,000 possible PIN codes because 10 x 10 x 10 x 10 = 10,000.** — With repetition allowed, each of the 4 positions has 10 choices (0-9), and choices are independent. Total = 10^4 = 10,000 codes (from 0000 to 9999).
A class has 10 students. How many ways can you choose a president, vice president, and secretary?	**There are 720 ways because 10 x 9 x 8 = 720.** — Since the positions are distinct (order matters) and one person can't hold multiple offices, this is a permutation: P(10,3) = 10 x 9 x 8 = 720.
Design a simple substitution cipher that is harder to break than the Caesar cipher. Explain why it's more secure.	**Use a random mapping where each letter maps to a different letter (not just a fixed shift). This has 26! (about 4 x 10^26) possible keys instead of just 25, making brute force impractical.** — A general substitution cipher assigns each letter to a random different letter: A→M, B→X, C→F, etc. There are 26! ≈ 4 x 10^26 possible mappings, making brute force impossible. However, frequency analysis (E is most common in English) can still crack it. This is why modern ciphers use much more sophisticated methods.
What is the Sieve of Eratosthenes, and how does it find prime numbers?	**The Sieve of Eratosthenes finds all primes up to a given number by systematically crossing out multiples of each prime, starting with 2.** — List numbers from 2 to N. Start with 2: it's prime, so cross out 4, 6, 8, 10... Next uncrossed number is 3: cross out 9, 15, 21... Next is 5: cross out 25, 35... The remaining uncrossed numbers are all prime. This 2,000+ year-old algorithm is still used in modern computing.
What is Fermat's Little Theorem, and how is it used in cryptography?	**Fermat's Little Theorem states that if p is prime and a is not divisible by p, then a^(p-1) mod p = 1. It's used in RSA to efficiently compute modular exponentiation.** — Fermat's Little Theorem: if p is prime and GCD(a,p) = 1, then a^(p-1) ≡ 1 (mod p). Example: 3^4 = 81, and 81 mod 5 = 1. In RSA, this theorem ensures that encryption followed by decryption returns the original message. It also enables efficient primality testing.
Explain what it means for a problem to be 'NP-complete' in simple terms.	**An NP-complete problem is one where no known fast algorithm exists, but if you're given a proposed solution, you can verify it quickly. If a fast algorithm is found for any NP-complete problem, it would solve all of them.** — NP-complete problems have two properties: (1) Solutions can be verified quickly (in polynomial time). (2) They are among the hardest problems in NP — every NP problem can be transformed into an NP-complete problem. The P vs NP question (are these problems actually solvable quickly?) is the biggest open problem in computer science, with a $1 million prize.
What is memoization, and how does it improve the recursive Fibonacci algorithm?	**Memoization stores previously computed results to avoid redundant calculations. It improves recursive Fibonacci from O(2^n) to O(n) by caching each Fibonacci value after computing it once.** — Naive recursive Fibonacci recomputes the same values exponentially: fib(50) would take trillions of calls. Memoization stores each result: when fib(k) is first computed, save it. If fib(k) is needed again, return the cached value in O(1). This reduces fib(n) from O(2^n) to O(n) — a massive improvement.

What is expected value, and how do you calculate it?	**Expected value is the average outcome you'd expect over many trials, calculated by multiplying each outcome by its probability and adding them together.** — Expected value (EV) is the long-run average. For a fair die: EV = (1)(1/6) + (2)(1/6) + (3)(1/6) + (4)(1/6) + (5)(1/6) + (6)(1/6) = 21/6 = 3.5. No single roll equals 3.5, but the average over many rolls approaches it.

15. Introduction to Sets

✂ Question (fold →)	Answer
A PE teacher has students line up by birthday (January first, December last). She can either use bubble sort (compare neighbors) or selection sort (find the earliest birthday and move them to the front). With 25 students, which algorithm would likely finish faster in real life, and why?	**Selection sort would likely be faster because students can scan the whole line visually to find the earliest birthday, rather than making many small swaps.** — While both algorithms have similar computational complexity ($O(n^2)$), selection sort is often faster in practice for humans because we can visually scan a group quickly. Selection sort also minimizes the number of actual moves — each person moves at most once. Bubble sort requires many small swaps, which is physically slow with people moving around.
In modular arithmetic, what is (7 + 9) mod 5?	**(7 + 9) mod 5 = 16 mod 5 = 1, because 16 divided by 5 gives remainder 1.** — Two methods: (1) Direct: 7 + 9 = 16, and 16 mod 5 = 1. (2) Reduce first: 7 mod 5 = 2, 9 mod 5 = 4, then (2 + 4) mod 5 = 6 mod 5 = 1. Both give the same answer. Note: the first distractor has the right idea but got 3 instead of 1 (arithmetic error demonstration).
Define the sequence T(n) recursively: T(1) = 3, T(n) = 2 x T(n-1) + 1. Find T(4).	**T(4) = 31, found by computing T(2) = 7, T(3) = 15, T(4) = 2 x 15 + 1 = 31.** — T(1) = 3. T(2) = 2 x 3 + 1 = 7. T(3) = 2 x 7 + 1 = 15. T(4) = 2 x 15 + 1 = 31. Each step doubles the previous value and adds 1. The sequence is 3, 7, 15, 31, 63, ... which equals 2^(n+1) - 1.
A game has a 1/6 chance of winning each round. What is the probability of losing 3 rounds in a row?	**The probability is (5/6)^3 = 125/216, which is about 57.9%.** — P(lose one round) = 1 - 1/6 = 5/6. Since rounds are independent, P(lose 3 in a row) = (5/6) x (5/6) x (5/6) = 125/216 ≈ 0.579 or about 57.9%.
Why is a recursive definition sometimes preferred over a closed-form formula?	**Recursive definitions are sometimes simpler to understand and write, especially when each step depends directly on the previous step.** — Recursive definitions model step-by-step processes naturally. The Fibonacci recursive rule (add the last two numbers) is easy to understand, while its closed-form (Binet's formula) involves the golden ratio and square root of 5. However, closed-form formulas are faster for computing the nth term directly.
In a class of 25 students, what is the approximate probability that at least two students share a birthday? (Assume 365 days in a year.)	**The probability is about 57%, which is surprisingly high despite having far fewer students than days.** — This is the Birthday Problem. It's easier to calculate the complement: P(no shared birthdays) = (365/365) x (364/365) x ... x (341/365). This gives about 0.43, so P(at least one shared birthday) ≈ 1 - 0.43 = 0.57 or 57%. The high probability surprises most people!
A programmer writes: 'if (age >= 13 OR hasParentPermission) AND accountActive.' Is this the right logic for 'users 13 and older can access the site, or younger users with parent permission, but only if their account is active'?	**Yes, this logic correctly captures the requirement. Users need an active account AND must either be 13+ or have parental permission.** — Let's verify: A 10-year-old with permission and active account: (false OR true) AND true = true ✓. A 15-year-old without permission, active: (true OR false) AND true = true ✓. A 14-year-old, active but deactivated account: (true OR false) AND false = false ✓. The parentheses correctly group the age/permission OR, then AND it with the account status requirement.
Why is P(A or B) = P(A) + P(B) - P(A	**We subtract P(A and B) to avoid counting the overlap where both events happen, which would otherwise be counted twice.** — When events overlap

and B) instead of just P(A) + P(B)?	(share outcomes), adding P(A) + P(B) counts the shared outcomes twice. Subtracting P(A and B) corrects this double-counting. This is the inclusion-exclusion principle from set theory applied to probability.
Compare three approaches to computing the sum 1 + 2 + 3 + ... + n: a loop, recursion, and a formula. Which is most efficient?	**The formula n(n+1)/2 is O(1) — instant regardless of n. A loop is O(n) — adds one number at a time. Recursion is also O(n) but uses O(n) stack space. The formula is most efficient.** — Formula: n(n+1)/2 — O(1) time, O(1) space. One multiplication and one division regardless of n. Loop: sum += i for i=1..n — O(n) time, O(1) space. Recursion: f(n) = n + f(n-1) — O(n) time, O(n) space (call stack). Gauss's formula is clearly superior, demonstrating that mathematical insight can dramatically outperform algorithmic brute force.
Design an efficient algorithm to determine if a word is a palindrome (reads the same forwards and backwards). What is its time complexity?	**Compare the first character with the last, second with second-to-last, and so on, moving inward. Stop when the pointers meet in the middle. Time complexity is O(n/2) = O(n).** — Two-pointer approach: set left = 0 (start), right = n-1 (end). While left < right: if word[left] ≠ word[right], return false. Otherwise, left++, right--. If all pairs match, return true. This is O(n) time, O(1) extra space. Alternative: reverse and compare is also O(n) but creates a copy (O(n) space).
What does it mean for something to be defined recursively?	**A recursive definition defines something in terms of itself, with a base case that stops the self-reference.** — Recursion defines something using a simpler version of itself. It needs two parts: a base case (the simplest case with a direct answer) and a recursive case (which reduces the problem to a simpler version). Without a base case, the definition would loop forever.
What are twin primes? Give three examples of twin prime pairs.	**Twin primes are pairs of primes that differ by exactly 2. Examples: (3, 5), (11, 13), and (17, 19).** — Twin primes are pairs of primes with a gap of 2. Examples: (3,5), (5,7), (11,13), (17,19), (29,31), (41,43). The Twin Prime Conjecture says there are infinitely many such pairs, but this remains unproven — one of math's great unsolved problems!
Explain why there are infinitely many prime numbers, using Euclid's argument.	**Assume there are finitely many primes. Multiply them all together and add 1. This new number isn't divisible by any known prime (remainder 1), so it must have a new prime factor — a contradiction.** — Euclid's proof (300 BCE): Suppose there are finitely many primes: p1, p2, ..., pk. Form N = p1 x p2 x ... x pk + 1. N gives remainder 1 when divided by any of our primes, so none of them divide N. But N must have a prime factor (Fundamental Theorem), and it's not in our list. Contradiction! So there must be infinitely many primes.
What does Big O notation measure?	**Big O notation describes how an algorithm's running time or space grows as the input size increases, focusing on the worst-case scenario.** — Big O notation classifies algorithms by their growth rate. O(1) is constant time (instant regardless of input). O(n) is linear (doubles with doubled input). $O(n^2)$ is quadratic (quadruples with doubled input). It focuses on worst-case behavior and ignores constants because growth rate matters most for large inputs.
Design a set-based system to organize your school's after-school clubs. Define at least three sets and show how union and intersection could help answer questions about student participation.	**You could define sets like Sports = {students in sports}, Arts = {students in art clubs}, and Academic = {students in academic clubs}, then use intersection to find students in multiple clubs and union to find all participating students.** — Sets are perfect for modeling group membership. Sports ∩ Arts finds multi-activity students who might have scheduling conflicts. Sports ∪ Arts ∪ Academic finds all students in at least one club. The complement (students NOT in any club) identifies those who might want recruiting. This is how databases, social networks, and analytics systems all use set theory.

Why is 4! (24) the number of ways to arrange 4 different books on a shelf?	**For the first position you have 4 choices, then 3 for the second, 2 for the third, and 1 for the last, giving 4 × 3 × 2 × 1 = 24.** — This is the multiplication principle in action. With 4 books: 4 choices for position 1, then 3 remaining for position 2, then 2 for position 3, and 1 for the last. The product 4 × 3 × 2 × 1 = 24 counts every possible arrangement. This is why n! counts the permutations of n distinct objects.
What is the Fundamental Theorem of Arithmetic?	**Every integer greater than 1 can be expressed as a unique product of prime numbers, regardless of the order of factors.** — The Fundamental Theorem of Arithmetic states that every integer > 1 has exactly one prime factorization (ignoring order). For example, $84 = 2 \times 2 \times 3 \times 7 = 2^2 \times 3 \times 7$, and there is no other way to express 84 as a product of primes. This uniqueness is the foundation of modern cryptography.
A pizza shop offers 3 sizes, 2 crust types, and 5 toppings. If you pick one of each, how many different single-topping pizzas are possible?	**There are 30 different single-topping pizzas, calculated as 3 × 2 × 5 = 30.** — Each pizza requires three independent choices: size (3 options), crust (2 options), and topping (5 options). The multiplication principle gives us 3 × 2 × 5 = 30 different combinations. This extends to any number of independent choices — just keep multiplying!
Decrypt the message 'KHOOR' using a Caesar cipher with shift 3.	**KHOOR decrypts to HELLO by shifting each letter back by 3: K→H, H→E, O→L, O→L, R→O.** — Decryption reverses the encryption: shift each letter back by 3. K(11)→H(8), H(8)→E(5), O(15)→L(12), O(15)→L(12), R(18)→O(15). KHOOR → HELLO. Caesar cipher encryption and decryption are inverse operations.
In a Venn diagram with two overlapping circles, where would you place an item that belongs to both Set A and Set B?	**An item belonging to both sets goes in the overlapping region where the two circles intersect.** — In a Venn diagram, the overlapping region (called the intersection) shows elements that belong to both sets simultaneously. Items only in Set A go in the left-only area, items only in Set B go in the right-only area, and shared items go in the middle overlap.
How does exclusive OR (XOR) differ from regular OR?	**XOR is true when exactly one input is true, but false when both are true, while regular OR is true when both are true.** — Regular OR (inclusive) is true when one or both inputs are true. XOR (exclusive OR) is true when exactly one input is true, but false when both are true. The menu example 'soup or salad' is XOR — you pick one. The security rule 'enter with a badge or sign in' is regular OR — you can do both. XOR is crucial in encryption and error detection.
An advertisement claims: 'Professional athletes use our shoes. If you use our shoes, you'll perform like a professional.' Is this a valid logical argument? Explain the flaw.	**This is not valid — it's the fallacy of affirming the consequent. Professional athletes using the shoes doesn't mean the shoes cause professional performance.** — The ad commits two logical errors: affirming the consequent (professionals use X, therefore X causes professionalism) and confusing correlation with causation. Professional athletes succeed due to years of training, genetics, coaching — not shoes. Recognizing these fallacies in advertising is a critical thinking skill that logic training develops.
Compare the efficiency of computing fib(20) recursively versus iteratively. Which is faster and why?	**Iterative is much faster because naive recursion recalculates the same values many times, while iteration computes each value once from left to right.** — Naive recursion computes fib(20) by calling fib(19) and fib(18), which both call fib(17), and so on. This creates an exponential explosion — fib(3) is computed 2,584 times! Iteration computes each Fibonacci number once in a loop, requiring only 20 steps. Memoization (caching results) can fix the recursive version.
	This is not always true because if A and B share elements, the union will

A student claims that if Set A has 4 elements and Set B has 6 elements, then A ∪ B must have exactly 10 elements. Is this always true? Explain why or why not.	**have fewer than 10 elements.** — The union has exactly 10 elements only if A and B are disjoint (share nothing). If they share elements, the union is smaller. For example, if A = {1, 2, 3, 4} and B = {3, 4, 5, 6, 7, 8}, the union is {1, 2, 3, 4, 5, 6, 7, 8} with 8 elements, not 10. The formula $\|A \cup B\| = \|A\| + \|B\| - \|A \cap B\|$ accounts for this.
If you have 3 shirts and 4 pairs of pants, how many different outfits can you make?	**You can make 12 different outfits, calculated as 3 × 4 = 12.** — The multiplication principle (also called the counting principle) states that if you have m choices for one decision and n choices for another independent decision, the total number of combined outcomes is m × n. With 3 shirts and 4 pants: 3 × 4 = 12 outfits.

16. Introduction to Sets

✂ Question (fold →)	Answer
You have 5 books to arrange on a shelf. If you sort them by title alphabetically, how many comparisons does it take using bubble sort in the worst case?	**Bubble sort needs at most 10 comparisons to sort 5 items, calculated as 4 + 3 + 2 + 1 = 10.** — Bubble sort compares adjacent elements and swaps them if needed. With 5 items, the first pass needs 4 comparisons, the second needs 3, the third needs 2, and the last needs 1. Total: 4 + 3 + 2 + 1 = 10. This pattern gives us n(n-1)/2 comparisons for n items.
Your friend argues: 'All dogs are animals. My cat is an animal. Therefore, my cat is a dog.' What logical error is your friend making?	**Your friend is committing the fallacy of affirming the consequent — just because all dogs are animals doesn't mean all animals are dogs.** — This is the fallacy of affirming the consequent. 'If dog, then animal' does NOT mean 'if animal, then dog.' Dogs are a subset of animals, but cats, birds, and fish are also animals. Confusing a statement with its converse is one of the most common logical errors in everyday reasoning, advertising, and even scientific claims.
Why is the Caesar cipher considered weak by modern standards?	**It is weak because there are only 25 possible shifts, so an attacker can try all of them quickly (brute force attack).** — With only 25 possible shifts (1 through 25), a brute-force attack simply tries each one. Additionally, letter frequency analysis can crack it: 'E' is the most common English letter, so the most frequent letter in the ciphertext likely represents 'E.' Modern ciphers use keys with trillions of possibilities.
Why are prime numbers important for modern internet security?	**Large prime numbers are the foundation of RSA encryption because multiplying two large primes is easy, but finding the original primes from their product is extremely difficult.** — RSA encryption relies on the asymmetry: multiplying two primes (like 7919 x 7907 = 62,607,133) is quick, but factoring the product back into primes is extremely slow for large numbers. RSA uses primes with hundreds of digits, where factoring would take millions of years even with modern computers.
What is an algorithm?	**An algorithm is a step-by-step set of instructions for solving a problem or completing a task, where each step is clear and unambiguous.** — An algorithm is a finite sequence of well-defined instructions for solving a class of problems. Algorithms predate computers: Euclid's algorithm (300 BCE) for finding the GCD is one of the oldest. Key properties: finite, definite (unambiguous), effective (each step can be done), and produces output.
What is the relationship between Set Notation and the broader field of Introduction to Sets?	**Set Notation and Introduction to Sets are deeply connected — Set Notation provides specific insights that contribute to our overall understanding of Introduction to Sets.** — Set Notation and Introduction to Sets are deeply connected — Set Notation provides specific insights that contribute to our overall understanding of Introduction to Sets. This is a fundamental concept in Introduction to Sets.
In a game, you spin a spinner divided into 5 equal sections (red, blue, green, yellow, purple) and flip a coin. What is the probability of landing on blue AND getting heads?	**The probability is 1/10 because (1/5) x (1/2) = 1/10.** — The spinner and coin flip are independent events. P(blue) = 1/5 and P(heads) = 1/2. For independent events, P(A and B) = P(A) x P(B) = 1/5 x 1/2 = 1/10.

Using the Sieve of Eratosthenes, find all prime numbers between 1 and 30.	**The primes between 1 and 30 are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 — there are 10 primes.** — Start: 2-30. Cross out multiples of 2 (4,6,8,...,30). Cross out multiples of 3 (9,15,21,27). Cross out multiples of 5 (25). Stop since $7^2 = 49 > 30$. Remaining: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29. Note: 1 is neither prime nor composite by definition.
What does it mean when two logical expressions are 'logically equivalent'?	**Two expressions are logically equivalent when they have identical truth tables — they produce the same output for every possible input combination.** — Logical equivalence means two expressions always agree. For example, 'NOT (P AND Q)' is logically equivalent to '(NOT P) OR (NOT Q)' — this is De Morgan's Law. You can verify by building truth tables for both and checking that every row matches. Logical equivalence is how mathematicians simplify complex expressions.
Two numbers are 'congruent modulo n' if they have the same remainder when divided by n. Are 38 and 14 congruent modulo 8?	**Yes, 38 and 14 are congruent modulo 8 because 38 mod 8 = 6 and 14 mod 8 = 6 — they have the same remainder.** — 38 mod 8 = 6 (since 38 = 4 x 8 + 6) and 14 mod 8 = 6 (since 14 = 1 x 8 + 6). Same remainder, so they are congruent modulo 8. Written: $38 \equiv 14 \pmod{8}$. Equivalently, 38 - 14 = 24, which is divisible by 8.
What is a Caesar cipher, and how would you encrypt the word 'MATH' with a shift of 3?	**A Caesar cipher shifts each letter by a fixed number. With shift 3, MATH becomes PDWK (M→P, A→D, T→W, H→K).** — The Caesar cipher replaces each letter with the one a fixed number of positions later in the alphabet. With shift 3: M→P, A→D, T→W, H→K. Julius Caesar used this to communicate with his generals. It's easily broken because there are only 25 possible shifts to try.
Describe how bubble sort works. What is its time complexity?	**Bubble sort repeatedly compares adjacent elements and swaps them if they're in the wrong order, making multiple passes until no swaps are needed. Its time complexity is $O(n^2)$.** — Bubble sort: compare pairs (1st & 2nd, 2nd & 3rd, ...), swap if out of order, repeat until sorted. Each pass moves the largest unsorted element to its correct position. It needs up to n-1 passes of up to n-1 comparisons each, giving $O(n^2)$. It's simple but slow for large lists.
A bag has 3 red marbles, 4 blue marbles, and 5 green marbles. If you draw one marble without looking, what is the probability of drawing a blue marble?	**The probability is 4/12 or 1/3, because there are 4 blue marbles out of 12 total.** — Total marbles = 3 + 4 + 5 = 12. Blue marbles = 4. Probability = favorable/total = 4/12 = 1/3.
What does 5! (five factorial) equal?	**Five factorial equals 120, calculated as 5 × 4 × 3 × 2 × 1 = 120.** — The factorial of a number n (written n!) is the product of all positive integers from 1 to n. So 5! = 5 × 4 × 3 × 2 × 1 = 120. Factorials grow incredibly fast: 10! = 3,628,800 and 20! is over 2 quintillion. Factorials are used to count permutations — there are n! ways to arrange n distinct objects.
What is modular arithmetic, and what does '17 mod 5' equal?	**Modular arithmetic is the math of remainders after division. 17 mod 5 = 2, because 17 divided by 5 is 3 remainder 2.** — Modular arithmetic finds the remainder after division. 17 ÷ 5 = 3 with remainder 2, so 17 mod 5 = 2. Clock arithmetic is modular: 17 hours on a 12-hour clock shows 5 (17 mod 12 = 5). Modular arithmetic is fundamental to cryptography.
How many ancestors	

do you have going back 10 generations (parents, grandparents, great-grandparents, etc.)?	**You have 2 + 4 + 8 + ... + 1024 = 2046 ancestors over 10 generations, which equals 2^11 - 2.** — Each generation doubles: 2, 4, 8, 16, ... 2^10 = 1024. The total is 2 + 4 + 8 + ... + 1024 = 2(2^10 - 1)/(2-1) = 2(1023) = 2046. This is a geometric series sum. (In reality, ancestors overlap due to intermarriage.)
Create a simple number-based encryption scheme using modular arithmetic. Encrypt the number 15 using your scheme and show how to decrypt it.	**Example scheme: Encrypt: C = (M x 3 + 7) mod 26. Encrypt 15: C = (15 x 3 + 7) mod 26 = 52 mod 26 = 0. Decrypt: find M where (M x 3 + 7) mod 26 = 0, giving M = 15.** — Encrypt: C = (M x k + s) mod n, where k is the multiplier and s is the shift. Decrypt: M = ((C - s) x k_inv) mod n, where k_inv is the modular inverse of k. For k=3, n=26: k_inv = 9 (because 3 x 9 = 27 ≡ 1 mod 26). Decrypt: M = ((0 - 7) x 9) mod 26 = (-63) mod 26 = 15.
What is a stable sorting algorithm, and why does stability matter?	**A stable sort preserves the relative order of elements with equal keys. It matters when sorting by multiple criteria — for example, sorting students first by name, then by grade, where students with the same grade stay in alphabetical order.** — Stability means equal elements maintain their original relative order. Example: sorting employees by department. If Alice and Bob are both in 'Engineering' and Alice appeared first originally, a stable sort keeps Alice before Bob. Merge sort and insertion sort are stable; selection sort and quicksort (typically) are not.
You need a 4-digit PIN using digits 0-9. How many PINs are possible if repetition is allowed?	**There are 10,000 possible PINs, calculated as 10 × 10 × 10 × 10 = 10,000.** — With repetition allowed, each of the 4 positions has 10 independent choices (0-9), giving 10^4 = 10,000 possible PINs (from 0000 to 9999). PINs starting with 0 are valid! Without repetition, it would be 10 × 9 × 8 × 7 = 5,040 PINs — nearly half as many.
How does a hash function work, and why can't you reverse it?	**A hash function converts any input into a fixed-size output (hash). It's a one-way function: many different inputs can produce the same hash, so you can't determine the original input from the hash alone.** — Hash functions map inputs of any size to a fixed-size output. They're one-way because information is lost: many inputs map to the same hash (collision). Your password is stored as a hash — when you log in, the system hashes your input and compares hashes, never storing or seeing your actual password.
Design an algorithm to find the second-largest number in a list. What is its time complexity?	**Track two variables: largest and second_largest. Scan through the list once, updating both as needed. Time complexity: O(n) since you examine each element once.** — Algorithm: Set largest = second_largest = -infinity. For each element x: if x > largest, set second_largest = largest, then largest = x. Else if x > second_largest (and x ≠ largest), set second_largest = x. This uses one pass (O(n)) and constant extra space (O(1)). Much faster than sorting (O(n log n)) for this specific task.
You have the set of colors {red, blue, green, yellow, purple}. Create a subset of primary colors from this set.	**The subset of primary colors is {red, blue, yellow}.** — The primary colors in traditional color theory are red, blue, and yellow. Since all three of these are in our original set, {red, blue, yellow} is a valid subset. Note that RGB (red, green, blue) are the primary colors of light, not paint — context matters in classification!
What is the golden ratio, and how does it relate to the Fibonacci sequence?	**The golden ratio (approximately 1.618) is the value that the ratio of consecutive Fibonacci numbers approaches as the numbers get larger.** — As Fibonacci numbers grow, the ratio F(n+1)/F(n) approaches φ ≈ 1.61803... The ratios are: 2/1=2, 3/2=1.5, 5/3≈1.667, 8/5=1.6, 13/8=1.625, 21/13≈1.615... converging to φ. The golden ratio appears in art, architecture, and nature.
What is the difference	**In permutations order matters, but in combinations order does not matter.** —

between a permutation and a combination?	Permutations count arrangements where order matters (ABC is different from BAC). Combinations count selections where order doesn't matter (choosing A, B, C is the same regardless of order).
If Set A = {apple, banana, cherry} and Set B = {banana, date, elderberry}, what is the union of A and B?	**The union of A and B is {apple, banana, cherry, date, elderberry}.** — The union of two sets (written A ∪ B) combines all elements from both sets, listing each element only once. Even though banana appears in both sets, it's listed just once in the union. Think of union as 'everything from either set.'

Keep going

You practiced **400 cards** from DiscreteQuest. Come back to the ones you missed — spaced-out practice makes them stick.

Keep exploring online

Play it now	Meet the cast	More free books
`https://spark-and-anvil.org/play/discretequest`	`https://spark-and-anvil.org/cast/discretequest`	`https://spark-and-anvil.org/books`

Scan a code with any phone camera, or type the address. Every app, story, and book is free, forever — no account, no tracking.