The Complete Optiver Assessment Prep Guide ��� Expanded Edition

THE COMPLETE OPTIVER ASSESSMENT PREP GUIDE — EXPANDED EDITION

SGP — Future Focus Trading & Research 2026

Beat The Odds · NumberLogic · Likelihood-list · Intervals · Orderbooks · Zap-N — with every idea and intuition explained

---

PART 0 — META-STRATEGY & MENTAL MATH

0.1 What these tests are really measuring (and why it changes how you take them)

Optiver is a market maker. A market maker continuously quotes two prices on financial instruments — a price they'll buy at and a price they'll sell at — and earns the gap between them, while trying not to get picked off by people who know something they don't. Strip that job down to its cognitive components and you get exactly these six tests:

- A market maker must **price uncertain outcomes in seconds** → Beat The Odds.
- They must **spot structure in streams of numbers** → NumberLogic.
- They rarely need exact probabilities, but constantly need to know **which of several scenarios is more likely** → Likelihood-list.
- Their quoted spread is literally a **confidence interval** around fair value: quote too tight and informed traders run them over; too wide and they do no business → Intervals.
- Their core profit engine is noticing when **two ways of buying the same thing have different prices** → Orderbooks.
- And all of it happens under **time pressure with constant context-switching** → Zap-N.

Why does this framing matter for you? Because it tells you the *grading philosophy*. A school exam rewards getting everything right. A trading-firm screen rewards behaving like a profitable trader: taking good bets, declining bad ones, being honest about uncertainty, and never freezing. Several of these tests literally encode that philosophy into the scoring (penalties for wrong answers, rewards for calibrated intervals, credit for correctly saying "no trade"). So "full marks" is not achieved by answering everything — it's achieved by maximizing expected score. Internalize this now; it recurs in every section.

0.2 The three golden rules, derived

Rule 1 — Your test-taking is itself an expected-value problem

Suppose a test awards +1 for a correct answer and −1 for a wrong one, with 0 for a skip (a common trading-test scheme — always check the rules shown on screen). You face a question with 4 options and no idea. Should you guess?

The expected score of a random guess is: EV = (1/4)(+1) + (3/4)(−1) = 0.25 − 0.75 = **−0.5**

A skip scores 0. So a blind guess is *worse than nothing* — it actively destroys half a point on average. The test is designed this way on purpose: in trading, a position you don't understand isn't neutral, it's dangerous, and the firm wants people who feel that instinctively.

Now suppose you can eliminate two options, leaving a 50/50 between the rest: EV = (1/2)(+1) + (1/2)(−1) = **0** — exactly breakeven, still not worth it (you spent time too).

And if you're 70% sure: EV = (0.7)(+1) + (0.3)(−1) = **+0.4** — now it's a good bet. Take it.

The intuition to carry: every question is a small trade. Your confidence is the probability; the scoring rule sets the payoff. The breakeven confidence for a (+1/−1) scheme is 50%; for (+1/−0.25) it's 20%; for (+1/0, no penalty) it's "always answer." Spend ten seconds at the start of each test reading the scoring rule, compute your personal breakeven once, and then apply it mechanically all the way through. This is worth more points than any single piece of mathematics in this guide.

Rule 2 — Why approximation beats precision here

Beat The Odds answer options are typically spaced like 0 / 0.1 / 0.2 / 0.5 / 1. Look at what that spacing means: the gap between adjacent options is enormous — often a factor of 2 or more. To pick the right option you don't need the third decimal; you need to know roughly which *zone* the answer lives in. Computing $(5/6)^4$ exactly takes a minute; knowing it's "a bit under a half" takes five seconds, and both lead to the same selected option.

This isn't laziness — it mirrors the job. A trader quoting a price doesn't solve an integral; they bound the value ("it's worth more than X because…, less than Y because…") and quote inside the bounds. Train yourself to produce bounds first, exact values only when the bounds straddle two options.

Rule 3 — Why time pressure is the real exam

A 90-second-per-question timer changes the optimal strategy in a way most candidates never consciously process: **every question is worth the same, but questions don't cost the same.** An easy question answered in 20 seconds and a brutal one answered in 90 both pay +1. So the skill being tested is not "can you solve hard problems" — it's "do you harvest cheap points fast and refuse to sink time into expensive ones." In tests that allow navigation (NumberLogic typically does), do a first pass collecting everything easy, then return. In tests that don't (Beat The Odds typically locks you in), set an internal tripwire: if at 40 seconds you have no method, switch from "solve" to "bound and pick" or skip.

There's a second, sneakier effect of time pressure: it amplifies *misreading*. Under stress, your eyes skip words like "not," "at least," "without replacement," "strictly." A large share of wrong answers on these tests are not math errors but reading errors. The fix is mechanical: as you read, subvocalize the quantifier words. It costs two seconds and saves entire questions.

0.3 Sitting logistics (the free points)

You can split the assessments across sittings. This is a gift — take it.

- **Do Zap-N first and freshest.** Reaction-time and working-memory scores degrade measurably with mental fatigue, and unlike the math tests, you cannot compensate with technique. Probability skills survive tiredness far better than reaction speed does.
- **Two to three tests per sitting, maximum.** Cognitive output drops nonlinearly after ~90 minutes of intense focus.
- **Mid-morning is the standard recommendation:** 1–2 hours after waking, after food, caffeine at your habitual dose ~30–45 minutes before.
- **Environment:** full screen, notifications off, phone in another room, pen and paper in hand (assume no calculator — these are mental-math tests by design), stable wired internet if possible.
- **Sleep the two nights before.** One bad night measurably slows reaction time; for Zap-N that's a direct score hit.

0.4 Mental math — what to memorize and *why each technique works*

Mental math is the substrate under everything else: a probability question that requires $(5/6)^4$ is trivial if 625 and 1296 surface instantly and impossible if they don't. Memorization isn't optional; but understanding why each shortcut works makes it stick and tells you when it applies.

The memory tables (drill 10 minutes daily until instant)

- **Squares to 30^2:** 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900
- **Cubes to 12^3:** 1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, 1331, 1728
- **Powers of 2 to 2^{12}:** 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096
- **Factorials to 8!:** 1, 2, 6, 24, 120, 720, 5040, 40320
- **Unit fractions:** 1/3 ≈ .333 · 1/6 ≈ .167 · 1/7 ≈ .143 · 1/8 = .125 · 1/9 ≈ .111 · 1/11 ≈ .091 · 1/12 ≈ .083 · 1/13 ≈ .077 · 1/16 = .0625
- **Primes to 100:** 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97
- **Triangular numbers:** 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105

Why these specific tables? Squares and cubes are the skeletons of NumberLogic sequences. Powers of 2 are coin-flip denominators. Factorials and the combination table are counting fuel. Unit fractions convert every dice/card probability to a decimal you can compare against the answer options. Triangular numbers are simultaneously a sequence family AND the "number of pairs" formula.

The techniques, with their mechanisms

Squaring numbers ending in 5. $75^2 = 5625$, computed as 7×8 = 56, append 25. *Why it works:* $(10a+5)^2 = 100a^2 + 100a + 25 = 100 \cdot a(a+1) + 25$. The a(a+1) is "digit times next digit"; the +25 is the appended tail. Works for any size: 115^2 → 11×12 = 132, append 25 → 13225.

Difference of squares for nearby numbers. $48 \times 52 = 50^2 - 2^2 = 2496$. *Why:* $(m-d)(m+d) = m^2 - d^2$. Whenever two factors straddle a round number symmetrically, jump to the round number's square and subtract the small square. 19 × 21 = 400 − 1. 37 × 43 = 1600 − 9.

Multiplying near 100. 97 × 96 = 9312, via deficits 3 and 4: left part 97−4 = 93 (or 96−3, same), right part 3×4 = 12. *Why:* $(100-a)(100-b) = 100(100-a-b) + ab$. The "cross subtraction" gives the hundreds; the product of deficits gives the tail. Same logic works above 100 with surpluses: 103 × 106 = 109 | 18 = 10918.

×5, ×25, ×50 as fraction moves. ×5 = ×10 ÷ 2; ×25 = ×100 ÷ 4; ×50 = ×100 ÷ 2. *Why:* 5 = 10/2, 25 = 100/4, 50 = 100/2 — replacing an awkward multiplier with a shift-and-halve. 84 × 25 = 8400 ÷ 4 = 2100. Division by 5: multiply by 2, divide by 10 (345 ÷ 5 = 690 ÷ 10 = 69).

Doubling–halving. 16 × 35 = 8 × 70 = 560. *Why:* multiplication is invariant when one factor doubles and the other halves. Keep trading until one factor is trivial. Most useful when one factor is even and the other ends in 5.

Percentage commutativity. 16% of 25 = 25% of 16 = 4. *Why:* a% of b = ab/100 = b% of a. Always flip to whichever side is the easy percentage.

Fraction ladder for probabilities. Convert fractions to decimals by anchoring on the unit-fraction table: 5/36 = 5 × (1/36) ≈ 5 × 0.028 = 0.14. 7/12 = 7 × 0.083 ≈ 0.58. This converts every "out of 36" dice answer and every "out of 52" card answer into the decimal the answer options are written in.

Estimating $(a/b)^n$. You'll constantly need things like $(5/6)^4$. Two routes: direct power tables ($5^4 = 625$, $6^4 = 1296$ → ≈ 0.48), or the exponential approximation $(5/6)^4 = (1 - 1/6)^4 \approx e^{(-4/6)} \approx e^{(-0.67)} \approx 0.51$. Both land you on the same answer option. The exponential route is explained fully in §1.9 — it's one of the most powerful ideas in the whole guide.

PART 1 — BEAT THE ODDS (Probability, 45 min, ~90 s/question)

Format: multiple-choice probability questions, hard per-question timer, coarse answer options (pick the *closest*), typically no returning to skipped questions, and typically penalties for wrong answers — so Rule 1 applies in full force.

1.1 What probability *is* (the model under every question)

Every probability question, no matter how dressed up, has the same skeleton: a **sample space** (the complete list of things that could happen) with a probability attached to each outcome, and an **event** (the subset of outcomes you care about). When all outcomes are equally likely — dice, cards, coins, "chosen at random" — probability collapses to pure counting:

P(event) = (number of favorable outcomes) / (total number of outcomes)

This sounds obvious but carries a profound practical consequence: **most probability errors are sample-space errors, not arithmetic errors.** When you get a question wrong it's usually because you counted the wrong universe — you forgot that (3,4) and (4,3) are different dice outcomes, or you kept outcomes the given information had eliminated. So the first move on every question is the same: *what is the full list of equally likely outcomes here, and which ones does the question's information rule out?* Get the universe right and the arithmetic is usually trivial.

A concrete illustration of why ordered outcomes matter: "two dice sum to 7" has six favorable outcomes — (1,6), (2,5), (3,4), (4,3), (5,2), (6,1) — out of 36 ordered pairs. A common error is to count unordered pairs ({1,6}, {2,5}, {3,4} → "3 out of 21") which gives a wrong answer because *unordered pairs are not equally likely*: {1,6} can happen two ways but {3,3} only one. The rule: keep outcomes at the level where symmetry guarantees equal likelihood — for dice and coins, that's the ordered level.

1.2 The five laws, each with its mechanism

Law 1 — Complement: P(not A) = 1 − P(A)

Mechanism: A and not-A split the sample space with no overlap and no gap, so their probabilities must sum to 1.

Why it's the single most used trick in this test: questions containing "**at least one**" describe an event that's a sprawling union (at least one six in four rolls = six on roll 1, OR roll 2, OR..., with messy overlaps). But its complement — "no sixes at all" — is one clean ANDed event: every roll individually avoids six, and independent ANDs just multiply. The complement converts an ugly union into a tidy product. **Reflex to build: see "at least one" → immediately write 1 − P(none).**

P(at least one six in 4 rolls) = $1 - (5/6)^4 = 1 - 625/1296 \approx 1 - 0.48 =$ **0.52**

The same reflex inverts: "P(none)" or "P(all fail)" questions are already in product form — just multiply.

Law 2 — Addition: P(A or B) = P(A) + P(B) − P(A and B)

Mechanism: adding P(A) and P(B) double-counts the outcomes that are in both, so subtract the overlap once. Picture two overlapping circles: area of the union = sum of areas minus the lens in the middle.

The recognition cue: the word "or." And the classic trap is forgetting the overlap term when the events *can* co-occur. P(king or heart) = 4/52 + 13/52 − 1/52 = 16/52 ≈ 0.31 — the king of hearts was counted in both groups and must be evicted once. Only when events are mutually exclusive (sum = 7 or sum = 11 — a roll can't be both) does the overlap vanish and pure addition apply.

Law 3 — Multiplication for independent events: P(A and B) = P(A)·P(B)

Mechanism: independence means B's chances are unaffected by whether A happened. Of the fraction of worlds where A happens, the *same* fraction P(B) of those also have B — a fraction of a fraction, hence multiplication.

Recognition: separate dice, separate flips, draws **with replacement**, unrelated people's birthdays. The trap is applying it to dependent things — consecutive card draws without replacement are NOT independent, which brings us to:

Law 4 — General multiplication: P(A and B) = P(A) · P(B | A)

Mechanism: this is just the definition of conditional probability rearranged. To get both A and B: first A must happen, then, *in the world where A already happened*, B must happen. The phrase "in the world where A happened" is exactly what P(B|A) means.

The workhorse application — sequential draws without replacement: P(both red from 5 red + 3 blue) = (5/8) × (4/7). After the first red leaves, the universe has shrunk to 7 balls of which 4 are red. **The mechanical habit: at each draw, update both the numerator and the denominator before multiplying.** Every "drawn without replacement" question is a chain of these shrinking fractions.

Law 5 — Total probability: P(B) = P(B|A)P(A) + P(B|not A)P(not A)

Mechanism: the worlds where B happens can be partitioned by whether A also happened; sum the two routes, each weighted by how likely that route is. It's a weighted average of B's probability across scenarios.

Recognition: two-stage problems — "a random bag is chosen, then a ball is drawn"; "a machine is selected, then an item is produced." Whenever a probability depends on an earlier hidden choice, total probability stitches the scenarios together. It's also the denominator of Bayes' theorem (§1.6), which is why it matters so much.

1.3 Counting — the engine room

Why counting is hard for humans (and the one question that fixes it)

Permutations vs combinations confuse people because the formulas look similar. The disambiguating question is physical, not algebraic: **"If I swapped two of the selected items, would I have a different outcome?"**

- Choosing a president and a treasurer: swapping them changes who holds which office → order matters → **permutation**: P(n,k) = n!/(n−k)!.
- Choosing a 2-person committee: swapping the members changes nothing → order doesn't matter → **combination**: C(n,k) = n!/(k!(n−k)!).

Why the formulas have the shape they do: building an ordered selection of k from n is a sequence of shrinking choices: n options, then n−1, then n−2, … (k factors) — that's n!/(n−k)!. If order doesn't actually matter, every unordered group of k has been counted once for each of its k! internal orderings, so divide by k!: that's the combination formula. Combinations are just permutations with the redundancy divided out. Once you see C(n,k) = (ordered count)/(orderings per group), you can re-derive anything you forget mid-test.

Memorize the small combinations (they appear constantly): C(4,2)=6 · C(5,2)=10 · C(5,3)=10 · C(6,2)=15 · C(6,3)=20 · C(7,2)=21 · C(7,3)=35 · C(8,2)=28 · C(8,4)=70 · C(10,2)=45 · C(10,3)=120 · C(52,2)=1326. Also the symmetry C(n,k) = C(n,n−k): choosing who's *in* is the same act as choosing who's *out* — which is why C(10,7) is just C(10,3).

"Number of pairs" = C(n,2) = n(n−1)/2. Handshakes, head-to-head matches, comparisons, possible two-card hands — any "how many pairs" question is this one formula. (Notice n(n−1)/2 is also the (n−1)th triangular number — the tables connect.)

The multiplication principle and the "glue" trick

Multiplication principle: a procedure done in stages with a, then b, then c options has a×b×c outcomes — provided the number of options at each stage doesn't depend on earlier choices. This single principle generates all of counting.

The glue (block) trick: "Alice and Bob must sit together in a row of 5." Treat the pair as one glued unit → 4 units arrange in 4! ways; the glue has 2 internal orders (AB/BA) → 4!×2 = 48 of 5! = 120 → P = 0.4. *Why it works:* gluing enforces adjacency by construction, and the two-stage count (arrange units, then arrange within the glue) uses the multiplication principle. **General insight: P(two specific people adjacent in a random row of n) = 2/n** — derive it once via glue (2·(n−1)!/n! = 2/n) and reuse forever.

Stars-and-bars (occasionally needed)

Distributing k identical items into n boxes: C(k+n−1, n−1) ways. *Mechanism:* lay the k items in a row as stars and insert n−1 bars to cut them into n groups; every arrangement of stars and bars is one distribution. Appears in "how many ways to split" questions.

1.4 Dice — the most common question source, fully unpacked

The two-dice sum table (commit to memory — it answers a third of dice questions on sight):

Sum	2	3	4	5	6	7	8	9	10	11	12
Ways (of 36)	1	2	3	4	5	6	5	4	3	2	1

Why this triangular shape: a sum of s can be made as (1, s−1), (2, s−2), … and the count grows by one per step until 7, where the dice faces start running out of room, after which it shrinks symmetrically. The symmetry around 7 exists because replacing each die's face f with 7−f is a bijection mapping sum s to sum 14−s — so P(s) = P(14−s) always. This symmetry is itself a tool: P(sum ≥ 10) must equal P(sum ≤ 4) without counting.

Max/min questions — the cumulative trick. "P(max = 4)" is hard directly (one die is 4 and the other ≤ 4, or both are 4 — overlap headaches) but trivial through cumulatives: P(max ≤ k) = $(k/6)^n$ for n dice — *every* die must be ≤ k, an AND of independents. Then P(max = k) = P(max ≤ k) − P(max ≤ k−1) — exactly-k is the thin shell between two nested "at most" events. *Why this is deep:* "convert exact to cumulative-difference" works everywhere in probability (it's how distribution functions work). For minimums, mirror it: P(min ≥ k) = $((7-k)/6)^n$.

Conditioning on dice information. "Given the sum is even, P(sum = 8)?" — conditioning means *physically shrink the universe to the allowed outcomes and recount*. Even sums: 1+3+5+5+3+1 = 18 outcomes; sum-8 contributes 5 → 5/18. Never compute conditionals by gut adjustment of the unconditional answer; always recount inside the restricted universe.

Repeated rolling — the geometric distribution. "Expected rolls until a six" = 6. *Two ways to see it:* 1. Self-consistency: E = 1 + (5/6)E — you always spend one roll; with probability 5/6 you're back where you started. Solve: E = 6. 2. Long-run frequency: sixes arrive once per 6 rolls on average, so the average gap is 6. Generalize: expected trials until an event of probability p is **1/p**. And P(first success on trial k) = $(1-p)^{k-1}p$ — failures then one success.

Races between events. "P(sum 7 before double-six)?" Ignore every roll that's neither — they don't affect the race. Among deciding rolls, the chance the race ends in a 7 is its share of the deciding probability: **P(A before B) = p_A/(p_A + p_B).** *Why ignoring non-deciding rolls is legitimate:* each roll is independent, so conditioned on "this roll decides," the relative odds of the two deciders are unchanged. This template kills craps-style questions, "first to appear" questions, and many game questions in one line.

1.5 Coins — binomials and the power of Pascal's rows

The structure: n fair flips → 2^n equally likely sequences. P(exactly k heads) = C(n,k)/2^n, because exactly C(n,k) sequences contain k heads (choose which positions are heads).

Memorize Pascal rows 4, 5, 6: n=4: 1 4 6 4 1 (/16) · n=5: 1 5 10 10 5 1 (/32) · n=6: 1 6 15 20 15 6 1 (/64) With these, every small-coin question is read off instantly: P(≥4 heads in 5) = (5+1)/32 ≈ 0.19. P(exactly 3 in 6) = 20/64 ≈ 0.31.

Why each row is built from the one above (Pascal's rule): a sequence of n flips with k heads either ends in heads (so the first n−1 flips have k−1 heads) or tails (first n−1 have k heads): C(n,k) = C(n−1,k−1) + C(n−1,k). You can rebuild a forgotten row in seconds.

Symmetry shortcuts. With an odd number of flips, ties are impossible, so P(more heads than tails) = exactly 1/2 — no computation. P(odd number of heads in any n ≥ 1 flips) = 1/2 (flip the last coin: it toggles parity, pairing every odd-heads sequence with an even-heads one). Train yourself to *look for the symmetry before counting* — it's the difference between 5 seconds and 60.

Streak anchors (calibration for "in a row" questions): 5 heads in a row = 1/32 ≈ 3%; 10 in a row = 1/1024 ≈ 0.1%. Each extra flip halves it. These anchors let you place any streak question on the answer ladder instantly.

Pattern races are NOT symmetric — the trap to respect. P(HH before HT) = 1/2 (after the first H, the very next flip decides). But P(HH before TH) = **1/4**: think about it — TH wins unless the sequence *starts* HH, because the moment any T ever appears, the next H completes TH before HH can ever form. Waiting times differ too: expected flips to see HH is 6; to see HT is only 4 (§8A.7 has the full machinery). The lesson: patterns with self-overlap behave differently; never assume two equal-length patterns are interchangeable. If a question races two patterns, trace the state logic explicitly.

1.6 Cards — and the symmetry principle that trivializes half of them

Deck facts: 52 cards = 4 suits × 13 ranks; 26 red, 26 black; 12 face cards (J, Q, K of each suit).

Instant probabilities: specific card 1/52 ≈ 0.019 · any named rank 4/52 = 1/13 ≈ 0.077 · second card matches first's rank ("a pair") 3/51 ≈ 0.059 · second matches first's suit 12/51 ≈ 0.235 · second matches first's color 25/51 ≈ 0.49.

The position-symmetry principle (the most elegant tool in the test)

Claim: in a shuffled deck about which you have **no information**, every position is equivalent. P(the 13th card is an ace) = P(the 1st card is an ace) = 4/52. P(the bottom card is a spade) = 1/4.

Why: a uniform shuffle treats all 52! orderings as equally likely; relabeling positions is a bijection on orderings, so no position is special. The aces don't know where they are.

Why this feels wrong (and why the feeling is the trap): intuition says "but 12 cards were dealt before it — doesn't that change things?" It would *if you saw them*. Information changes probabilities; mere position does not. The moment the question reveals card values ("the first card was an ace"), symmetry breaks and you recount: P(second is an ace | first was) = 3/51. **Decision rule: revealed information → recount the shrunken universe; no revelation → symmetry answers instantly.**

A beautiful consequence: P(second card is a king), with the first card drawn but *unseen*, is exactly 4/52 — averaging over what the unseen first card might be (total probability: (4/52)(3/51) + (48/52)(4/51) = 4/52) collapses back to the symmetric answer. The unseen card may as well still be in the deck.

Two standard computations, both ways

Exactly one heart in two draws. Combination route: C(13,1)C(39,1)/C(52,2) = 507/1326 ≈ 0.38. Sequential route: 2 × (13/52)(39/51) ≈ 0.38 — the factor 2 because heart-first and heart-second are two disjoint orderings with equal probability. Knowing both routes lets you cross-check in seconds; agreement = locked answer.

Gap symmetry for "expected position of first ace": the 4 aces cut the other 48 cards into 5 gaps which are, by symmetry, the same size on average: 48/5 = 9.6 cards per gap, so the first ace sits at expected position 9.6 + 1 = **10.6**. General formula: (n+1)/(k+1). No summation needed — symmetry again.

1.7 Conditional probability & Bayes — the deepest idea in the test

What conditioning really is

P(A|B) = P(A and B)/P(B). The formula is dry; the picture is everything. **Conditioning is universe replacement.** When you learn that B happened, the outcomes outside B stop existing for you. Your new universe is B, and the probability of A is now just A's share *of B* — which is exactly what the ratio computes: how much of B is also A.

Every conditional-probability blunder in history is the same blunder: answering about the old universe instead of the new one. The "two children" problem makes this vivid:

A family has two children; at least one is a boy. P(both are boys)? Original universe: BB, BG, GB, GG — four equally likely. "At least one boy" deletes GG. New universe: BB, BG, GB. Both-boys is 1 of these 3 → **1/3**. *Variant: "the older child is a boy."* This deletes GB and GG. New universe: BB, BG → **1/2**.

Why do the answers differ when the sentences sound so similar? Because the two pieces of information delete *different amounts of universe*. "At least one boy" is weak information (kills one outcome of four); "the older is a boy" is strong (kills two). Weaker information leaves more competing possibilities, so the target event keeps a smaller share. **Read conditioning statements with lawyer-level precision — the exam writers choose their words to exploit exactly this.**

Bayes' theorem in the form you'll actually use

The textbook form ($P(A|B) = P(B|A)P(A)/P(B)$) is correct and almost useless under a 90-second timer. Use **natural frequencies** instead: imagine 1,000 cases and push them through the story.

The canonical disease problem: 1% prevalence; test 95% sensitive (catches the sick), 90% specific (clears the healthy). You test positive. P(actually sick)?

Out of 1,000 people: - 10 are sick → 9.5 test positive (true positives) - 990 are healthy → 10% err → 99 test positive (false positives) - Positive testers: 9.5 + 99 = 108.5, of whom sick = 9.5 → P = 9.5/108.5 ≈ **0.09**

The intuition that must become permanent: your gut says ~95% because the test is "95% accurate." But accuracy operates on two wildly different-sized groups. The healthy group is 99× larger than the sick group, so even its small 10% error rate produces a *bigger pile* of false positives than the sick group's true positives. **When the base rate is small, the false positives from the huge healthy majority swamp the true positives from the tiny sick minority.** A positive test doesn't tell you "you're probably sick"; it tells you "you've moved from 1% to 9%" — a 9× update, but on a tiny starting point.

The frequency method generalizes to every Bayes question: pick a convenient population, split by the prior, apply the conditional rates to each branch, then read off the answer as (target branch)/(all branches matching the evidence). It's the same theorem with the algebra dissolved into counting.

Sequential evidence — Bayes as repeated multiplication

Two coins: one fair, one double-headed. Pick one at random; it flips heads. P(it's the trick coin)? Weights: trick = ½ × 1 = ½; fair = ½ × ½ = ¼. Posterior = (½)/(½ + ¼) = **2/3**. *Both flips heads?* trick = ½ × 1; fair = ½ × ¼ → posterior = 4/5. *Three heads?* → 8/9.

See the pattern: every additional head multiplies the *odds* in favor of the trick coin by 2 (the ratio of how well each hypothesis predicts a head: 1 vs ½). This is the **odds form of Bayes**: posterior odds = prior odds × likelihood ratio, applied once per piece of evidence. Under time pressure, tracking odds (2:1, then 4:1, then 8:1) is faster than tracking probabilities, and converting back at the end is one step: odds a:b → probability a/(a+b).

Monty Hall — and the general principle it teaches

Three doors, one prize. You pick a door; the host, *who knows where the prize is*, opens a different door revealing a goat, and offers a switch. Switching wins with probability **2/3**.

The cleanest way to see it: your original pick was right 1/3 of the time and wrong 2/3 of the time — nothing the host does changes that, because he can *always* perform his reveal regardless of your pick. In every world where your pick was wrong (2/3 of them), the host's hand is forced: the remaining closed door is the prize. Switching converts "initially wrong" into "win." So switch-win-probability = P(initially wrong) = 2/3.

The principle underneath: the host's reveal carries information **because his behavior is constrained** — he'll never open your door or the prize door. If instead a door opened at random and happened to show a goat, the answer would be 1/2: random events that *could have* gone otherwise carry different information than forced events. When an exam question involves someone revealing something, always ask: *could they have revealed otherwise? what were they required to do?* That determines how much the revelation shifts probabilities.

1.8 Expected value — the trader's native language

What EV is and why traders worship it

$E[X] = \Sigma$ (value × probability) — the probability-weighted average outcome. Its claim to authority is the **law of large numbers**: over many repetitions, your average realized result converges to the EV. A trader makes thousands of small bets; the noise cancels and the EV is what remains. That's why the only question a trader asks of a bet is not "will it win?" but "is it +EV, and by how much?"

Fair price = the EV. A game paying the face of one die roll has E = (1+…+6)/6 = 3.5: pay less than $3.50 and you profit on average; more and you bleed. When a test asks "fair price" or "how much would you pay," it's asking for the EV, full stop.

Breakeven probability — the instant bet-evaluation tool: a bet paying b:1 breaks even when $p \times b = (1-p) \times 1$, i.e., $p^* = 1/(b+1)$. Pays 3:1 → breakeven 25%. If your estimated probability beats p^*, the bet is +EV. This converts every "should you take this bet" question into one comparison.

Linearity of expectation — the superpower

E[X + Y] = E[X] + E[Y], always — even when X and Y are dependent. No independence requirement whatsoever. This is the most powerful and least intuitive tool in the section.

Why it's true: expectation is a weighted sum, and sums can be reorganized term by term; correlations affect the *spread* of X+Y, never its average.

Why it's a superpower: it lets you compute expectations of horribly entangled quantities by decomposing into trivial indicator pieces. The hat problem: 100 hats returned at random, expected number of people with their own hat? The events are wildly dependent (if 99 people got their own, the last must too). Irrelevant! Each person individually has a 1/100 chance; E = 100 × 1/100 = **1**. Done.

The general recipe (indicator method): expected count of anything = Σ over items of P(that item qualifies). Expected sixes in 30 rolls = 30 × 1/6 = 5. Expected distinct faces in 3 rolls = 6 × P(a given face appears) = $6 \times (1 - (5/6)^3) \approx 2.5$. Expected pairs among n people sharing a birthday = C(n,2)/365. Whenever a question says "expected number of…," your first thought should be: *can I write this as a sum of per-item probabilities?* The answer is almost always yes.

The re-roll option (option value made concrete)

Roll a die; you may re-roll once and must keep the second roll. Value? Work backwards. The re-roll, if used, is worth 3.5. So you keep any first roll beating 3.5 — keep 4, 5, 6; re-roll 1, 2, 3. Value = ½ × (average of 4,5,6) + ½ × 3.5 = ½ × 5 + ½ × 3.5 = **4.25**. *With two re-rolls:* the continuation is now worth 4.25, so on the first roll keep only 5 or 6: value = (2/6)(5.5) + (4/6)(4.25) = **4.67**.

The embedded lessons: (1) optionality has value — each added re-roll raises the game's worth even though you might not use it; (2) the threshold for "keep" is the value of what you'd be giving up (the continuation), not some absolute standard; (3) solve sequential decisions backwards from the end. These three ideas are the kernel of all option pricing, and the exam loves dressing them in dice.

Two famous expectations worth owning

Coupon collector: expected die rolls to see all six faces = 6 × (1 + ½ + ⅓ + ¼ + ⅕ + ⅙) ≈ **14.7**. *Why:* while you've seen j faces, the chance a roll shows a new one is (6−j)/6, so that phase lasts 6/(6−j) rolls on average; sum the phases. The first new face is instant; the last takes 6 rolls on average — the end dominates the wait.

Geometric mean wait: expected trials to an event of chance p = 1/p (derived in §1.4). Pairs with: P(still waiting after k trials) = $(1-p)^k$.

1.9 Geometric probability — when outcomes are continuous

When a point is chosen "uniformly at random" from a line segment, region, or interval of time, probability = (favorable measure)/(total measure) — favorable length over total length, or area over area. Counting becomes measuring; everything else is unchanged.

Dartboard: circle inscribed in a square. P(random point in the square lands in the circle) = $\pi r^2/(2r)^2$ = **π/4 ≈ 0.785**. (Worth memorizing as a constant — it recurs.)

The meeting problem (the template for two continuous variables): two people arrive uniformly within an hour; each waits 15 minutes. P(they meet)? Draw the unit square of arrival times (x, y). They meet when |x − y| ≤ ¼ — a diagonal band. The complement is two clean corner triangles, each with legs ¾: missed area = $(3/4)^2$ = 9/16. **P(meet) = 7/16 ≈ 0.44.** *The transferable method:* any problem with two independent uniform quantities → draw the square, shade the condition, compute areas. Bands, triangles, and circular arcs cover essentially every exam case. The general formula for this template: $P = 1 - (1 - w/T)^2$.

Known results worth carrying: break a stick at two uniform points — P(pieces form a triangle) = **1/4**. Three random points on a circle all on one semicircle: **3/4** (in general n points: $n/2^{n-1}$). Two uniforms on [0,1]: E|X−Y| = **1/3**, P(|X−Y| < d) = $1 - (1-d)^2$.

1.10 The approximation toolkit — derived, not decreed

The exponential bridge: $(1-p)^n \approx e^{-np}$

Where it comes from: ln(1−p) ≈ −p for small p (the first term of the Taylor series). So $(1-p)^n = e^{n \cdot \ln(1-p)} \approx e^{-np}$. The error is tiny when p ≤ 0.2 and acceptable even at p = 1/3.

Why it's transformative: it converts powers (slow to compute mentally) into a single product np and a lookup. Memorize four e-values: $e^{-0.5} \approx 0.61$, $e^{-0.7} \approx 0.50$, $e^{-1} \approx 0.37$, $e^{-2.3} \approx 0.10$. Then:

P(at least one success in n tries) ≈ $1 - e^{-np}$, where np = expected number of successes.

- np ≈ 0.7 → P(at least one) ≈ 50%
- np ≈ 1 → ≈ 63%
- np ≈ 2.3 → ≈ 90%
- np ≈ 3 → ≈ 95%

De Méré's bet (at least one double-six in 24 rolls): np = 24/36 = 0.67 → ≈ $1 - e^{-0.67}$ ≈ 49%. The birthday problem runs on the same engine with np = C(n,2)/365. One idea, dozens of questions.

Bounds as answers

The union bound: P(A or B or C…) ≤ P(A) + P(B) + P(C)… — the sum double-counts overlaps, so it can only overshoot. Combined with the trivial lower bound (the union is at least as likely as its biggest member), you get a sandwich: max single ≤ P(union) ≤

sum. With answer options spaced a factor of 2 apart, the sandwich frequently contains exactly one option — answer found without computing anything exactly.

The normal approximation for large counts

A count of n fair coin flips has mean n/2 and standard deviation **$\sqrt{n}/2$** (100 flips → SD 5; 400 → SD 10). Then use the z-ladder: beyond 1 SD one-sided ≈ 16%, beyond 2 SD ≈ 2.3%, beyond 3 SD ≈ 0.1%. "P(≥ 60 heads in 100)" → 60 is 2 SD up → ≈ 2–3%. *Why $\sqrt{n}$:* variances add across independent flips (np(1−p) = n/4), and SD is the square root. The deep fact (central limit theorem) is that sums of many independent pieces always look bell-shaped — which is why this one ladder serves every large-count question.

The decision flowchart for any Beat The Odds question

1. **Symmetry scan (5 s):** does symmetry give the answer outright (position symmetry, sum symmetry, parity, equal-by-relabeling)?
2. **Phrase triggers:** "at least one" → complement · "expected number" → linearity/indicators · "before" → race template · "given/if" → restrict the universe and recount · "fair price" → EV.
3. **Count or multiply (30 s):** small universe → count it; sequential draws → chain of shrinking fractions; independent repetitions → powers.
4. **40-second tripwire:** no clean route? Bound it (union bound + e-approximation), pick the option inside the bounds.
5. **Sanity direction-check (5 s):** does the answer move the right way? (Conditioning on supporting evidence raises probability; small base rates stay smallish; adding requirements lowers probability.)
6. **Skip if confidence < breakeven** (Rule 1). A disciplined skip is a correct answer.

1.11 Rapid-fire practice set (20 questions, 90 s each)

1. Two dice: P(sum ≥ 9)?
2. Flip 6 coins: P(exactly 3 heads)?
3. Draw 2 cards: P(both same rank)?
4. Roll a die twice: P(second strictly greater than first)?
5. 3 red, 2 blue balls drawn out one by one: P(last ball is blue)?
6. A die rolled 3 times: P(strictly increasing rolls)?
7. Two children, the younger is a girl: P(both girls)?
8. Urn 4 white, 6 black, draw 3: P(all black)?
9. Expected heads in 12 flips?
10. Roulette (37 slots, 18 red): EV of $10 on red at even money?
11. Two dice: P(product odd)?
12. Random arrangement of A,B,C,D,E: P(starts with A)?
13. P(at least one double-six in 24 rolls)?
14. Coins: 2 fair + 1 double-headed; random coin flips heads. P(double-headed)?
15. Random integer 1–100: P(divisible by 3 or 5)?
16. 6 people in a row: P(two specific people occupy the two ends)?
17. Expected distinct faces in 3 die rolls?
18. P(a 5-card hand contains all four aces)?
19. Two dice, given at least one 6: P(sum = 8)?
20. Game: $100 with probability 0.04, costs $5. EV?

Answers & routes: 1. 10/36 ≈ **0.28** · 2. 20/64 ≈ **0.31** (Pascal row 6) · 3. 3/51 ≈ **0.06** · 4. (1 − 1/6)/2 = **5/12 ≈ 0.42** (symmetry: ties aside, greater/less are equally likely) · 5. **2/5** (position symmetry — last ball is a uniformly random ball) · 6. $C(6,3)/6^3$ = 20/216 ≈ **0.09** (choose 3 distinct values; only 1 of their 6 orders ascends) · 7. **1/2** (the older child is unconstrained) · 8. (6/10)(5/9)(4/8) = **1/6** · 9. **6** · 10. (18−19)/37 × 10 ≈ **−$0.27** · 11. $(1/2)^2$ = **0.25** (both must be odd) · 12. **1/5** (symmetry) · 13. $1 − (35/36)^{24}$ ≈ 1 − e^(−2/3) ≈ **0.49** · 14. weights 1 vs ½+½ → **1/2** · 15. (33+20−6)/100 = **0.47** (inclusion–exclusion) · 16. 2!·4!/6! = **1/15 ≈ 0.07** · 17. $6(1 − (5/6)^3)$ ≈ **2.5** (indicators) · 18. 48/C(52,5) ≈ **0.00002 → "≈0"** · 19. 11 outcomes contain a 6; {(2,6),(6,2)} → **2/11 ≈ 0.18** · 20. 4 − 5 = **−$1, decline**

PART 2 — NUMBERLOGIC (Sequences, 25 min)

2.1 What pattern recognition actually is (and why an algorithm beats staring)

A sequence question hands you 5–7 numbers and asks for the generating rule. The naive approach — stare and hope — fails under time pressure because the space of plausible rules is large and your attention is hijacked by whatever rule you saw on the *previous* question (a real cognitive effect: priming). The professional approach is a fixed diagnostic ladder run in the same order every time. Two reasons this wins: (1) the ladder is ordered by frequency — the most common families are tested first, so expected time-to-solution is minimized; (2) a fixed procedure is immune to priming and to panic.

One principle governs the whole part: **a rule must fit every given term with zero exceptions.** A rule that fits four of five terms is not "close" — it is wrong, and patched rules ("×2+1 except the third step") are essentially never the intent. The moment a hypothesis fails one term, discard it entirely and step down the ladder. Sunk-cost attachment to a half-working rule is the single biggest time sink in this test.

2.2 The diagnostic ladder, with the reasoning behind each rung

Rung 1 — First differences. Write the gaps. Why first: differencing is the universal linearizer — it strips one layer of structure off whatever is there. Constant gaps → arithmetic, done. Gaps forming their own visible pattern (odds, squares, doubling) → the sequence is one integral up from a simple one; predict the next gap and add. **Constant second differences → quadratic** (an n^2 family) — this is worth understanding: just as constant 1st differences mean a linear rule (slope), constant 2nd differences mean a quadratic, with the constant equal to 2a for rule an^2+bn+c. You rarely need the explicit formula — extending the difference pyramid one cell answers the question.

Rung 2 — Ratios. Divide consecutive terms. Constant ratio → geometric. Ratios forming a pattern (×2, ×3, ×4…) → factorial-flavored. Why second: multiplication is the second-most-common engine, and ratios detect it as cleanly as differences detect addition. *Tell-tale for choosing rung 1 vs 2 at a glance:* roughly even spacing → think differences; accelerating, snowballing growth → think ratios. Fractional/decimal terms (40.5, 10.125) scream a non-integer ratio like ×1.5.

Rung 3 — The alternation test. Read every second term: positions 1,3,5,… as one subsequence and 2,4,6,… as another. If each is simple alone, you have **interleaved sequences** — NumberLogic's signature trap. Why it fools people: your eye insists on reading left to right as one stream. The trigger to check: differences that jump around wildly *in sign or size* (+9, −7, +11, −5…) — oscillation is the fingerprint of two braided strands. Also try a +a/×b alternating single-strand rule (3, 5, 10, 12, 24 → +2, ×2 repeating).

Rung 4 — Affine two-step rules: ×m ± k. Test ×2+1, ×2−1, ×3+1, ×3−2, ÷2+2… Practical detector: compute the rough ratio of late consecutive terms — 67/22 ≈ 3.05 whispers "×3 and a small correction"; then solve the correction from one pair and verify on all the others. Why late terms: as numbers grow, the additive correction shrinks relative to the multiplier, so late ratios reveal m most clearly.

Rung 5 — Neighbor recursion. Is each term built from the previous two (sum: Fibonacci-like; product; difference)? Detector: the sequence's own earlier terms seem to "reappear" inside the differences — indeed for Fibonacci-type rules the differences literally reproduce the sequence shifted by one. Any seeds with the sum rule make Fibonacci-like growth (ratio drifting toward 1.618 — a useful fingerprint).

Rung 6 — Famous sequences, possibly shifted. Squares, cubes, primes, triangular numbers, powers of 2, factorials — and their ±1/±2 disguises (n^2+1, 2^n-1). This is where the memory tables pay off: 624 doesn't look like anything until 625 is reflexive. *Heuristics:* a leading 0 → shifted famous sequence ($(n-1)^3$…); large "ugly" terms (343, 1331, 3125) are almost always pure powers; all-even terms → factor out 2 and re-run the ladder on the halves (4, 6, 10, 14, 22 → 2 × primes).

Rung 7 — Digit operations. Next term built from the *digits* of the previous: digit sums, digit products, reversals, look-and-say. Trigger: a sudden collapse in magnitude (86 → 14 → 5) or a repeated term (digit operations hit fixed points — single-digit numbers map to themselves).

Rung 8 — Position rules. Term = f(n) directly: $n(n+1)$, $n^2(n+1)$, n^3-n, n^n. Often discoverable via rung 1's difference pyramid anyway, but factor-shaped values (6, 24, 60, 120, 210 = $n(n+1)(n+2)$) can be seen directly.

2.3 Worked taxonomy — 30 sequences with reasoning

A. Differences

A1. 7, 12, 17, 22 → +5 → **27** **A2.** 3, 4, 7, 12, 19 → gaps 1, 3, 5, 7 (odds) → +9 → **28** **A3.** 2, 3, 6, 11, 18, 27 → gaps 1, 3, 5, 7, 9 → **38** **A4.** 1, 2, 6, 15, 31 → gaps 1, 4, 9, 16 (squares) → +25 → **56** **A5.** 5, 6, 9, 14, 21, 30 → gaps odd numbers → +11 → **41** *Reasoning pattern: the gaps themselves are a famous sequence — two layers, each simple.*

B. Ratios

B1. 3, 6, 12, 24 → ×2 → **48** **B2.** 2, 6, 18, 54 → ×3 → **162** **B3.** 64, 96, 144, 216 → ×1.5 → **324** *(96/64 = 1.5 — always divide to find a non-obvious ratio)* **B4.** 1, 2, 6, 24, 120 → ratios 2, 3, 4, 5 → ×6 → **720** *(growing ratios = factorials)* **B5.** 4, 6, 12, 30, 90 → ratios 1.5, 2, 2.5, 3 → ×3.5 → **315**

C. Affine two-step

C1. 3, 7, 15, 31, 63 → ×2+1 → **127** *(also 2^n-1 — two readings, one answer: a good sign)* **C2.** 2, 5, 11, 23, 47 → ×2+1 → **95** **C3.** 5, 9, 17, 33, 65 → ×2−1 → **129** *(equivalently 2^n+1)* **C4.** 1, 4, 13, 40, 121 → ×3+1 → **364** **C5.** 100, 52, 28, 16, 10 → ÷2+2 → **7** *(decreasing toward the rule's fixed point 4 — affine rules converge to $k/(1-m)$ when $|m|<1$, a fingerprint of ÷-rules)*

D. Interleaved

D1. 1, 10, 3, 20, 5, 30 → strands 1,3,5,(7) and 10,20,30 → **7** **D2.** 2, 100, 4, 90, 8, 80, 16 → doubling / −10 → **70** **D3.** 5, 8, 10, 16, 20, 32 → both strands ×2 → **40** **D4.** 3, 4, 7, 8, 11, 12 → single strand, +1/+3 alternating → **15** **D5.** 1, 1, 2, 4, 3, 9, 4, 16 → pairs (n, n^2) → **5**

E. Recursion

E1. 1, 1, 2, 3, 5, 8, 13 → Fibonacci → **21** **E2.** 2, 3, 5, 8, 13, 21 → Fibonacci rule, offset seeds → **34** **E3.** 1, 2, 2, 4, 8, 32 → product of previous two → **256** **E4.** 10, 7, 3, 4, −1, 5 → $a_n = a_{n-2} - a_{n-1}$ → **−6** *(negative terms appearing mid-sequence point to subtraction recursions)* **E5.** 1, 2, 3, 6, 11, 20, 37 → sum of previous three → **68**

F. Famous, shifted

F1. 0, 3, 8, 15, 24 → n^2-1 → **35** **F2.** 2, 5, 10, 17, 26 → n^2+1 → **37** **F3.** 2, 3, 5, 7, 11, 13 → primes → **17** **F4.** 1, 3, 6, 10, 15 → triangular → **21** **F5.** 1, 8, 27, 64 → cubes → **125** **F6.** 0, 6, 24, 60, 120 → n^3-n → **210**

G. Digits

G1. 86, 14, 5, 5 → digit sums → **5** **G2.** 12, 21, 13, 31, 14 → reversal pairs → **41** **G3.** 99, 81, 8, 8 → digit products → **8** **G4.** 1, 11, 21, 1211, 111221 → look-and-say → **312211**

H. Mixed/hard

H1. 4, 9, 25, 49, 121 → squares *of primes* → **169** *(two famous sequences composed — if the squares' roots are themselves patterned, chase the roots)* **H2.** 6, 24, 60, 120, 210 → $n(n+1)(n+2)$ → **336** **H3.** 1, 4, 27, 256 → n^n → **3125** **H4.** 3, 5, 9, 17, 33 → 2^n+1 → **65** **H5.** 2, 12, 36, 80, 150 → $n^2(n+1)$ → **252** *(or pure difference-pyramid: gaps 10,24,44,70; 2nd diffs 14,20,26; 3rd diffs constant 6 → extend)*

2.4 Strategy, scoring, and the psychology of the test

- **Harvest pass first.** NumberLogic typically allows navigation. Pass 1: answer everything solvable inside ~40 s, flag the rest. Pass 2: return to flags. This guarantees the cheap points are banked before time dies on an expensive question.
- **The 40-second tripwire** (same as Beat The Odds): no hypothesis by 40 s → flag and move. Discipline here is worth several questions.
- **Verify on ALL terms, predict, move on.** Verification takes 5 seconds and catches the "fits-most-terms" traps that are deliberately planted.
- **Two independent routes agreeing = certainty.** Many sequences (C1, H5) admit both a positional formula and a difference reading; if both give the same next term, answer with full confidence.
- **Scoring:** typically no penalty for blanks and points per correct answer; if so, in the final 30 seconds fill every remaining blank with your best structural guess (e.g., extend the apparent trend). Confirm the scheme on screen first — if wrong answers cost points, leave true unknowns blank.

PART 3 — LIKELIHOOD-LIST (Ranking outcomes, ~25 min)

3.1 Why ranking is its own skill (and which mental bugs it exploits)

You're given a scenario and several outcomes to sort from most to least likely, under a per-question timer. The deep design insight: **comparing probabilities is much cheaper than computing them** — and the test is checking whether you know that. A candidate who computes four exact probabilities runs out of time; a candidate who sorts three of them by pure structure and computes only the one genuine near-tie finishes in half the time with the same accuracy.

The test simultaneously probes a famous catalogue of human biases. Knowing them by name inoculates you:

- **Conjunction fallacy:** vivid, detailed scenarios *feel* likelier than sparse ones, but every added detail mathematically lowers probability. P(A and B) ≤ P(A), with no exceptions, ever. "Bank teller and feminist activist" cannot beat "bank teller" — yet most untrained people rank it higher when a story makes the conjunction vivid. **Specificity always reduces probability.**
- **Availability bias:** dramatic outcomes (plane crash, lottery win) get ranked too high because they come to mind easily. Rank by counting and base rates, never by imaginability.
- **Representativeness:** the sequence HTHTH looks "more random" than HHHHH, but every exact 5-flip sequence is exactly 1/32. Patterns of equal length are equally likely as *exact sequences* — though not as *categories* (there are many sequences with 3 heads, only one with 5).

3.2 The six ranking principles, with mechanisms

P1 — Supersets beat subsets. If outcome X logically contains outcome Y (every Y-world is an X-world), then P(X) ≥ P(Y) — set containment is probability's hardest constraint. Scan for containment first: it sorts options with zero arithmetic. ("At least one head" ⊇ "exactly one head" ⊇ "heads on flip 1 only.")

P2 — More ways → more likely. With equally likely elementary outcomes, the comparison is just a count of ways. Sum 7 (6 ways) beats sum 10 (3 ways) — no division needed since both share the /36.

P3 — The threshold chain. P(at least k) ≥ P(exactly k) ≥ P(exactly at an extreme). "At least" sums multiple "exactly" terms, so it dominates each.

P4 — Each added independent requirement multiplies by something < 1. A chain of three conditions sits below any chain of two of them. This is P1's quantitative engine.

P5 — The reference ladder. Carry anchor probabilities and hang each option on the ladder: 1/2 (coin) · 1/3 · 1/4 · 1/6 (die face) · 1/13 (named rank) · ~1/30 (specific dice pair, 5-flip streak) · 1/52 · ~1/100 · ~1/1000 (10-flip streak). Sorting then needs only *which rung*, not the exact value.

P6 — Spend computation only on genuine near-ties. Sort the clear ones structurally; identify the one adjacent pair within ~20% of each other; compute that single comparison by counting ways. This is the entire time-management strategy of the section.

3.3 Worked rankings

3.3.1 *Two dice. Rank: (a) sum = 7, (b) doubles, (c) sum ≥ 10, (d) both dice ≥ 5.* Counts of 36: a = 6, b = 6, c = 6, d = 4 (2×2). **a = b = c > d.** The ranking needed zero division — counts suffice (P2).

3.3.2 *Five flips. Rank: (a) exactly 3 heads, (b) at least 4 heads, (c) the exact string HTHTH, (d) first two flips heads.* a = 10/32 ≈ .31 · d = 8/32 = .25 · b = 6/32 ≈ .19 · c = 1/32 ≈ .03. **a > d > b > c.** Note (c): an exact string is always $1/2^n$ regardless of how "balanced" it looks (representativeness trap).

3.3.3 *One card. Rank: (a) red, (b) a heart, (c) a face card, (d) queen of hearts, (e) heart or spade.* a = e = 26/52 > b = 13/52 > c = 12/52 > d = 1/52. The only computation worth doing: b vs c, settled by counting 13 vs 12. Everything else is structural.

3.3.4 *A random adult. Rank: (a) plays tennis, (b) plays tennis and is over 50, (c) is over 50, (d) plays tennis, is over 50, and plays weekly.* Structure (P1/P4): a > b > d and c > b > d. The only judgment call is a vs c — base rates: over-50 (~35–40%) dwarfs tennis (~5–10%). **c > a > b > d.** Notice how far structure alone carried the answer before any worldly knowledge was needed — that's the standard division of labor.

3.3.5 *Urn 50 red/50 blue, 10 draws with replacement. Rank: (a) exactly 5 red, (b) 10 red, (c) at least 6 red, (d) first draw red.* d = 0.5 · c ≈ (1 − P(exactly 5))/2 ≈ (1 − .246)/2 ≈ .38 · a = 252/1024 ≈ .246 · b ≈ .001. **d > c > a > b.** Anchor worth keeping: P(exactly half in 2n flips) ≈ 0.8/√n → for n = 5, ≈ 0.25 ✓. And the c-trick: by symmetry the non-tie probability splits evenly between "more red" and "more blue."

PART 4 — INTERVALS (Estimation & calibration, ~15 min)

4.1 What calibration means, precisely

You'll be asked to estimate quantities and wrap each estimate in a lower and upper bound. The scoring rewards intervals that contain the truth and are tight. This is a **calibration** test: when you say you're 90% sure, are you right 90% of the time?

Here's the uncomfortable, extremely well-replicated finding from decades of research: untrained people asked for 90% confidence intervals capture the truth only **40–60%** of the time. Not because they can't estimate — because their *intervals* are systematically far too narrow. Everyone's. Yours too, until trained. The bias is called overconfidence, and it has a known mechanism: when you estimate, your mind anchors on a point value and then adjusts outward "enough" — but the adjustment process stops too early, because each step away from the anchor feels increasingly unjustified, even though your actual uncertainty extends much further.

Why does Optiver care so much? Because a market maker's bid–ask spread *is* a confidence interval around fair value, posted with real money behind it. Quote too tight relative to your true uncertainty and informed traders systematically pick you off (the "miss" penalty, in cash). Quote wider than your uncertainty requires and you do no trades (the "width" penalty). A miscalibrated trader bleeds from one side or the other; a calibrated one survives. The test scoring mirrors this exactly: misses are penalized much more than width.

Therefore the single most valuable instruction in this entire guide: form your instinctive interval, then widen it — roughly double the width. It will feel cowardly. That feeling is the bias talking. Measured against reality, the doubled interval is the honest one.

4.2 Multiplicative thinking — why bounds belong in log-space

For counting/measuring questions ("how many X"), uncertainty is naturally multiplicative, not additive. If your estimate is 100,000 you are not "±20,000" — you are "within a factor of 2 or 3." The right interval is **[E/k, E×k]**, not [E−x, E+x].

Why: such quantities arise as products of factors (population × rate × frequency…), and uncertainty in a product compounds multiplicatively — being 2× off on one factor scales the whole answer by 2×. Additive intervals around big numbers are either uselessly narrow on the low side or nonsensically allow negatives. Log-space is also why your *point* estimate should be a geometric midpoint: unsure between 10,000 and 1,000,000? The honest center is $\sqrt{(10^4 \times 10^6)} = 100{,}000$ — not the arithmetic average 505,000, which secretly leans 5× toward the top.

Choosing k honestly: domain you know → k ≈ 1.5–2 · moderately familiar → k ≈ 3 · genuinely guessing the order of magnitude → k ≈ 5–10. And asymmetry: quantities bounded below by zero have long right tails, so push the upper bound out further than the lower (interval [E/2, E×3] is often more honest than [E/3, E×3]… the principle: more room above).

4.3 Fermi decomposition — why breaking problems up *reduces* error

The engine for "impossible" estimates: factor the unknown into 3–6 pieces you can each guess within a factor of 2–3, then multiply.

Why decomposition helps rather than compounding error: your factor errors are roughly independent and unbiased in log-space — some high, some low — so they partially **cancel** in the product. One direct gut guess has the full variance of your ignorance; a product of five half-informed factors has its errors diversified. It's the same mathematics as portfolio diversification. (The discipline that makes it work: estimate each factor *honestly on its own*, never steering factors toward an answer you secretly want.)

The five standard decompositions — most Fermi questions are one of these shapes: 1. **Population × rate × frequency** — "how many X-doers / X-events per day" 2. **Stock = flow × lifetime** — things currently in a system = arrival rate × average stay (works for airplanes aloft, students enrolled, cars on a road) 3. **Big measure ÷ measure per unit** — areas, volumes, packing 4. **Capacity × utilization × time** — throughput questions 5. **Customers × frequency × price** — revenue questions

Worked decompositions

Piano tuners in Chicago (the classic, with commentary): Chicago ≈ 3M people → ≈ 1.2M households (≈2.5 people/household — a number worth owning). Pianos: ~1 household in 20 → 60k, plus institutions → ~70k. Tuned ~1×/year → 70k tunings/yr. A tuner does ~3/day × 250 working days ≈ 750/yr. Tuners ≈ 70,000/750 ≈ **~90; interval [30, 300]**. Note where the uncertainty lives: the 1-in-20 piano rate is the shakiest factor (could be 1-in-10 or 1-in-40), so it alone justifies k ≈ 3.

Flights airborne worldwide right now (stock = flow × lifetime): ~100k commercial flights/day; average duration ~2 h → airborne fraction 2/24. Stock ≈ 100k × 2/24 ≈ **8–10k; interval [5k, 20k]**. The structural insight: "how many right now" is *always* flow × (duration/period).

Golf balls in a school bus (measure ÷ measure): Bus interior ~10 × 2.5 × 2.5 ≈ 60 m³, usable ~50. Ball: d ≈ 4.3 cm → $\sim 4\times10^{-5}$ m³; sphere packing ≈ 64%. $50 \times 0.64/4\times10^{-5}$ ≈ **800k; interval [300k, 2M]**.

MRT rides in Singapore per day (population × rate): 6M residents; transit-dominant city → ~half ride on a given day × 2 rides ≈ **3–3.5M; interval [2M, 5M]** (actual ≈ 3.5M — the decomposition lands on it).

When two routes agree, tighten. Estimate Changi's daily passengers via the hub anchor (~60M/yr ÷ 365 ≈ 165k) AND via flights (~1,000/day × ~150 pax ≈ 150k): agreement from independent routes is genuine evidence, and *only then* is a tight interval earned. This is the precise sense in which narrow intervals must be paid for with evidence.

4.4 Anchor numbers (your estimation vocabulary)

Populations: World 8B · China 1.4B · India 1.45B · USA 340M · Indonesia 280M · EU 450M · Japan 124M · UK 68M · **Singapore 6M** · Tokyo metro 37M · a "major city" 1–10M · households ≈ population/2.5 **Geography:** Earth circumference 40,000 km · surface 510M km² (71% ocean) · USA 9.8M km² · Singapore 730 km² · Everest 8.8 km · ocean max depth ~11 km **Human:** heart ~70 bpm ≈ 100k beats/day · ~20k breaths/day · life ≈ 80 yr ≈ 30,000 days ≈ $2.5×10^9$ s · walk 5 km/h · 2,000 kcal/day **Time:** 1 yr ≈ $3.15×10^7$ s ("$π×10^7$") · 1 day = 86,400 s · work-year ≈ 2,000 h **Physics/engineering:** light $3×10^8$ m/s · sound 343 m/s · 1 m³ water = 1,000 kg = 1 tonne · car ~1.5 t · loaded jumbo ~400 t · cruise altitude ~10–11 km · jet ~900 km/h · air density 1.2 kg/m³ **Economy:** world GDP ~$110T · US ~$29T · China ~$19T · Singapore ~$530B · 1 oz = 28.3 g **Daily life:** smartphones in use ~5B · world vehicles ~1.5B · a person eats ~1,000 kg/yr · supermarket ~40,000 SKUs

4.5 Calibration training protocol (it genuinely works in days)

1. Take 20 trivia quantities (the workbook has 60 with answers); write 90% intervals.
2. Score hits. First attempt is typically 8–14/20. Don't be discouraged — that's the universal starting point.
3. Mechanically widen until you genuinely hit 17–19/20 on fresh sets. The skill being built is not knowledge — it's an honest mapping from "how unsure I feel" to "how wide I write."
4. Track *which side* you miss on. Systematic low misses on big-world quantities (distances, populations) are the classic signature; correct the bias, not just the width.
5. In-test mantra: **"Would I be genuinely shocked if the truth were outside this range?"** Not mildly surprised — shocked. If not, widen. Never let the interval exclude any value you can't rule out with an articulable reason.

PART 5 — ORDERBOOKS (Arbitrage spotting, ~15 min)

5.1 The vocabulary, and the asymmetry everything depends on

An order book for an item shows two standing prices: the **bid** (the highest price someone is committed to *buying* at — therefore the price **you can sell at**) and the **ask** (the lowest price someone will *sell* at — the price **you can buy at**). The ask exceeds the bid; the gap is the **spread**.

Burn in the direction-mapping, because every error in this test is ultimately a direction error: **you buy at asks (the higher number), you sell at bids (the lower number).** You always transact at the *worse* of the two prices from your perspective — that's what it means to demand immediacy. A useful mnemonic: the market is a shop; the shop sells to you at its ask (retail price) and buys from you at its bid (trade-in price), and the shop's margin is the spread.

Arbitrage = a set of trades that leaves you with no items (flat) and positive cash, guaranteed. The test for any candidate combination is one line: **Σ(sale proceeds at bids) − Σ(purchase costs at asks) > 0, with item quantities netting to exactly zero.**

Why "flat" is part of the definition: leftover inventory is risk, not profit — its value could move. True arbitrage is riskless precisely because nothing remains to fluctuate.

5.2 Pattern 1 — The direct cross

If, anywhere, a bid ≥ an ask for the *same item* (e.g., one venue asks 99 while another bids 101), buy at the ask, sell at the bid, pocket the difference. In a healthy market this never persists — which is exactly why spotting it instantly is the test's first checkpoint. Scan time: ~5 seconds per screen.

5.3 Pattern 2 — Basket vs components (the core pattern)

A bundle and its parts are two routes to the same goods. If the routes are priced inconsistently *after crossing the spreads*, profit exists. Both directions must be checked:

- **Buy parts, sell bundle:** profit = bundle bid − Σ(component asks)
- **Buy bundle, sell parts:** profit = Σ(component bids) − bundle ask

Worked: Apple 10/11, Banana 5/6, Basket (1+1) 18/19. Direction 1: parts cost 11+6 = 17; basket bid 18 → **+1 ✓.** Direction 2: basket ask 19 vs parts bids 15 → −4 ✗. The directions are not mirror images — each crosses *different* spreads — so one can work while the other badly fails. **Checking both is non-negotiable**, and with coefficients (a bundle of 2A+1B), multiply quantities through: cost = 2×(A's ask) + 1×(B's ask).

Why "the mids look mispriced" is not a trade: suppose X mid 10, Y mid 5, bundle mid 17 — the bundle looks 2 rich. Execute it: sell bundle at bid 16, buy parts at asks 11+6 = 17 → **−1**. The apparent edge was smaller than the spreads you must cross to harvest it. **Profit is computed bid-vs-ask, never mid-vs-mid** — this is the single most realistic trap in the test, because it's the daily reality of the job.

5.4 Pattern 3 — Multi-bundle netting (the vector method)

When no single basket-vs-parts works, a *combination* of bundles may replicate another instrument. The systematic tool: write every instrument as a vector of item quantities plus a cash flow, and search for a combination whose items sum to zero with positive cash.

Worked: AB 14/15 · BC 9/10 · ABC 24/25 · B 2/3. Buy AB and BC: holdings A+2B+C, cash −25. Sell ABC: holdings B, cash −1. Sell B at 2: flat, cash **+1** ✓. Bookkeeping discipline: two columns — items and cash. An arbitrage is items = 0, cash > 0. With ≤5 instruments, testing each replication identity in both directions takes under a minute. Common identities: AB + C vs ABC · AB + BC vs ABC + B · AB + CD vs ABCD.

5.5 Pattern 4 — Conversion loops (triangles)

Quotes like "1 A converts to 2 B," "1 B to 3 C," "1 C to 0.18 A" form a loop. **Multiply the rates around the loop: product > 1 → running the loop multiplies your holdings; product < 1 → that direction loses,** and the reverse direction would profit *only if reverse legs are actually quoted* — never assume a quote is invertible. (2.0 × 1.5 × 0.36 = 1.08 → +8% per cycle; at 0.33 the product is 0.99 → no trade.) *Why multiplication:* each conversion scales your quantity by its rate; consecutive scalings compose by multiplication, so the loop's net effect is the product. Real-world name: triangular FX arbitrage.

5.6 The most important answer: "no arbitrage"

Some screens contain **no opportunity**, and saying so is the correct answer. No-arb conditions: every item's bid < ask; every bundle satisfies Σ(component asks) ≥ bundle bid AND bundle ask ≥ Σ(component bids); every loop's product ≤ 1 in all executable directions. The deeper point: a trader who *manufactures* trades where no edge exists is the most expensive kind of employee. The test deliberately includes no-trade screens to find candidates who can sit on their hands. Forcing a trade on a clean screen is scored as exactly what it is — a loss.

5.7 The per-screen routine (drill until ~60 seconds total)

1. Direct crosses? (any bid ≥ ask on the same item) — 5 s.
2. Every bundle, both directions: Σ component asks vs bundle bid; bundle ask vs Σ component bids — 20 s.
3. Netting: can bundle ± bundle replicate something quoted? Vector check — 20 s.
4. Loops: multiply rates around each executable cycle — 10 s.
5. Found a trade → compute per-unit profit, then size to the **smallest leg's available volume** (each leg caps the whole structure; after trading, the used quotes are gone — re-scan before assuming a second round).
6. Nothing → answer "no opportunity" with confidence.

PART 6 — ZAP-N (Neuro games)

6.1 What's being measured and what the games look like

Short games measuring raw cognitive throughput: **processing speed** (respond fast), **working memory** (hold and reproduce digit/position strings, forward and backward), **task switching** (the rule changes by cue: blue frame → judge odd/even, red frame → judge vowel/consonant), **inhibition** (Stroop-style: respond to the relevant attribute while a salient conflicting cue screams the wrong answer), **rapid comparison** (are two symbol strings identical?), and sometimes **risk calibration** (balloon-pump games: each pump adds reward, a pop loses it all).

Why a trading firm measures this: live markets are a Stroop test with money — relevant signals arrive mixed with salient irrelevant ones, rules change regime without warning, and several positions must be held in mind at once. Zap-N is the lab version.

6.2 What's trainable (more than you'd think, at the margins that matter)

Raw IQ won't move in a week; the *losses* from unfamiliarity, poor tactics, and poor state absolutely will — and they're worth 10–30% of score.

- **Familiarity:** play each game type until the rules are automatic, because the first 20 seconds of confusion in a 2-minute game is an unrecoverable 15% of it. Free Stroop/N-back/digit-span games online suffice.
- **Digit span technique — chunking:** group digits in 3s/4s with a rhythm ("314 – 159 – 26"); working memory holds ~4 chunks regardless of chunk size, so bigger chunks = longer spans. Backward recall: visualize the digits written out, read the image right-to-left. Trained spans: 7–9 forward, 5–7 backward.
- **Switching technique:** the cost is in *letting go* of the old rule. Subvocalize the active rule in one word at each cue ("color… color… SHAPE"). Anticipate the cue location with your eyes.
- **Inhibition technique:** deliberately run at ~95% of max speed. Accuracy typically weights more than raw speed, error streaks compound (an error rattles you into the next error), and the steady-95% candidate beats the frantic-80% one. After any error: one breath, slow 10% for five trials, resume.
- **Risk games:** the optimal policy is consistent moderate risk — neither max-greed (pops erase everything) nor max-caution (tiny banked rewards). If the game gives pop feedback, estimate the pop threshold from early balloons and stop ~2/3 of the way to it.
- **State (worth more than everything above):** 7–9 h sleep both prior nights (reaction time degrades 5–10% sleep-deprived — catastrophic in a speed test) · habitual caffeine 30–45 min before · Zap-N *first* in the sitting · 2 minutes of any fast-twitch warm-up game immediately before · read every instruction screen twice and always take offered practice rounds.

PART 7 — THE 7-DAY PLAN (deadline June 18)

Daily skeleton (2.5–3.5 h): 15 min mental-math drill → 60–90 min topic of the day (this guide) → 30 min timed workbook sets → 15 min calibration (20 intervals) → 15 min one neuro game.

- **Day 1:** Part 0 + §§1.1–1.6 (laws, counting, dice, coins, cards). Workbook Sets A–B untimed.
- **Day 2:** §§1.7–1.11 (Bayes, EV, geometric, approximations). Sets A–B **timed at 90 s/question**; Sets C–D untimed.
- **Day 3:** Part 2 in full. Workbook Blocks 1–5 timed at 60 s; Sets C–D timed.
- **Day 4:** Parts 3–4. Likelihood L1–L10; first full calibration scoring (30 questions); memorize anchor table.
- **Day 5:** Part 5; drill the per-screen routine on all 24 OB problems until both-direction checks are sub-20-seconds. Zap-N familiarization. Workbook Sets E–F timed.
- **Day 6:** Full mock day — the whole battery, two sittings, realistic timing (Sets G–H, Blocks 6–10, L11–L20, Calibration II, OB13–24). Log every miss by *cause*: knowledge / speed / misread / calibration.
- **Day 7:** Review only the miss-log and memory tables. Nothing new. Early night.

Final-hour checklist

☐ Tables fluent (squares, cubes, powers, C(n,k), two-dice sums, Pascal rows) ☐ “At least one” → complement, reflexively ☐ Bayes via 1,000-person frequencies ☐ Breakeven confidence computed from the on-screen scoring rule; skip below it ☐ Intervals widened to genuine-shock width, log-space ☐ Orderbooks: both directions on every bundle; mids are not prices; “no trade” is an answer ☐ Zap-N first, slept, warmed up ☐ Quantifier words subvocalized while reading ☐ Pen, paper, water

You’re not trying to be perfect. You’re trying to be calibrated, fast, and +EV — which is the actual job.

PART 8 — ADVANCED PROBLEM SETS (Harder questions, full solutions)

This part raises the difficulty to the top end of what these assessments throw at strong candidates. Every problem has a complete solution, including the *fast* route you'd actually use inside a 90-second window.

8A. Beat The Odds — Hard set (16 problems)

Problem 8A.1 — The birthday problem

A room has 23 people. P(at least two share a birthday)?

Full solution. Complement: P(all distinct) = (365/365)(364/365)(363/365)⋯(343/365). Take logs or use the approximation P(all distinct) ≈ $e^{-C(n,2)/365}$. With n = 23: C(23,2) = 253, so P(all distinct) ≈ $e^{-253/365} = e^{-0.693}$ ≈ 0.50. **Answer ≈ 0.50. 90-second route:** memorize the landmark — 23 people → 50%, 30 → 71%, 50 → 97%, 60 → 99.4%. Also memorize the tool: *P(no collision among n items in d slots) ≈ $e^{-n^2/2d}$* — it solves every "shared birthday/hash collision" variant instantly.

Problem 8A.2 — Expected rolls until two consecutive sixes

Roll a die repeatedly. Expected number of rolls until you see two sixes in a row?

Full solution. Let E = expected rolls from scratch, E_1 = expected additional rolls given the last roll was a six. From scratch: $E = 1 + (1/6)E_1 + (5/6)E$. After one six: $E_1 = 1 + (1/6)(0) + (5/6)E$. From the first equation: $(1/6)E = 1 + (1/6)E_1 \rightarrow E = 6 + E_1$. Substitute: $E_1 = 1 + (5/6)(6 + E_1) = 6 + (5/6)E_1 \rightarrow (1/6)E_1 = 6 \rightarrow E_1 = 36$. So **E = 42. Pattern to memorize:** expected wait for a p-probability event twice in a row = $1/p + 1/p^2$. For p = 1/6: 6 + 36 = 42. (Three in a row: 6 + 36 + 216 = 258.)

Problem 8A.3 — Race between sums (dice)

Roll two dice repeatedly. P(a sum of 7 appears before a sum of 8)?

Full solution. Only rolls landing 7 or 8 matter; all others are ignored ("condition on the deciding roll"). P(7) = 6/36, P(8) = 5/36. P(7 first) = (6/36)/(6/36 + 5/36) = **6/11 ≈ 0.545. Template:** in any race between events with per-trial probabilities p and q, P(p wins) = p/(p+q). This kills a whole family of "which comes first" questions in 10 seconds.

Problem 8A.4 — Gambler's ruin (fair game)

You have $3 and bet $1 on fair coin flips; you stop at $0 or $10. P(you reach $10)?

Full solution. For a fair game this is a martingale: your expected final wealth equals your starting wealth. Final wealth is 10 with probability w, 0 otherwise: 10w = 3 → **w = 0.3. General rule:** fair game, start i, target N → P(success) = i/N. Expected number of bets = i(N−i) = 3×7 = 21 (also worth memorizing).

Problem 8A.5 — Socks in the dark

A drawer has 10 distinct pairs of socks (20 socks). You grab 4 at random. P(at least one matching pair)?

Full solution. Complement: all 4 socks from different pairs. P(no pair) = (20/20) × (18/19) × (16/18) × (14/17). Cancel: = (16 × 14)/(19 × 17) = 224/323 ≈ 0.693. **Answer = 1 − 0.693 ≈ 0.31. Method note:** for "no two alike" draws, multiply sequential survival probabilities — at each step count how many remaining cards/socks *avoid* a match.

Problem 8A.6 — The defective-machine Bayes

Machines M1, M2, M3 make 30%, 50%, 20% of output with defect rates 2%, 3%, 5%. A random item is defective. P(it came from M3)?

Full solution. Per 1000 items: M1 → 300 × 2% = 6 defectives; M2 → 500 × 3% = 15; M3 → 200 × 5% = 10. Total = 31. P(M3 | defective) = 10/31 ≈ **0.32. Note the trap:** M3 has the *highest defect rate* but is not the most likely source (M2 is, at 15/31 ≈ 0.48) — production share fights defect rate. Always multiply rate × share before comparing.

Problem 8A.7 — Waiting for patterns: HH vs HT

Expected number of fair coin flips until you first see HH? Until HT?

Full solution (HH). States: start (S), "just saw H" ($\bar{H}$). $E_S = 1 + ½E_{\bar{H}} + ½E_S$; $E_{\bar{H}} = 1 + ½(0) + ½E_S$. From first: $½E_S = 1 + ½E_{\bar{H}} \rightarrow E_S = 2 + E_{\bar{H}}$. Substitute: $E_{\bar{H}} = 1 + ½(2 + E_{\bar{H}}) \rightarrow ½E_{\bar{H}} = 2 \rightarrow E_{\bar{H}} = 4 \rightarrow$ **E_S = 6. (HT):** once you see your first H, you just wait for a T (expected 2 more flips); expected wait for first H is 2. **Total = 4. The paradox to internalize:** both patterns are equally likely in any two flips, but HH takes longer on average because a failed attempt (HT when hunting HH) destroys your progress, while hunting HT, an "extra H" keeps your progress alive. Overlapping structure → longer waits.

Problem 8A.8 — Optimal stopping with a die

You may roll a die repeatedly. Each roll costs $1. You may stop at any time and collect the face value of your last roll. What's the *value of the game and the optimal strategy?*

Full solution. Let V = value of the game (before paying for a roll). You roll (pay 1), see face f, and keep it if f ≥ continuation value V, else roll again. Guess "stop on 4, 5, 6": V = −1 + (3/6)(average of 4,5,6) + (3/6)V = −1 + (1/2)(5) + (1/2)V → (1/2)V = 1.5 → **V = 3.** Consistency check: you stop when f > 3 and are exactly indifferent at f = 3 (since V = 3) — the guess is self-consistent. **Strategy: re-roll on 1–3, stop on 4–6; game is worth $3. General method for stop/continue questions:** guess a threshold, solve the linear equation, check the threshold is consistent with the value you got.

Problem 8A.9 — The ballot problem

Candidate A receives 6 votes, B receives 4, counted in random order. P(A is strictly ahead throughout the entire count)?

Full solution. Ballot theorem: P = (a − b)/(a + b) = (6−4)/(6+4) = **0.2.** This is worth knowing cold; the test can phrase it as profits/losses or goals scored. (The related "A never behind, ties allowed" version has probability (a+1−b)/(a+1).)

Problem 8A.10 — Normal approximation under time pressure

Flip a fair coin 100 times. Roughly P(at least 60 heads)?

Full solution. Mean = 50, SD = √(100 × ½ × ½) = 5. Sixty heads is 2 SD above the mean (with continuity correction, z = (59.5−50)/5 = 1.9). Tail beyond ~2 SD ≈ 2.5%. **Answer ≈ 0.02–0.03 → pick the option nearest 0.02/0.05. Memorize the z-ladder:** beyond 1 SD one-sided ≈ 16%; beyond 1.5 ≈ 7%; beyond 2 ≈ 2.3%; beyond 3 ≈ 0.1%. And SD of coin counts = √n/2 (so 100 flips → 5, 400 flips → 10).

Problem 8A.11 — All four aces together

Shuffle a deck. P(all four aces end up in four consecutive positions)?

Full solution. Think of choosing the 4 ace positions: C(52,4) equally likely position-sets = 270,725. Consecutive runs: 49 of them (starting at positions 1–49). P = 49/270,725 ≈ **0.00018 → "≈ 0". Speed lesson:** when the answer options include 0, recognize "essentially impossible" structurally (one tiny favorable family vs a combinatorial explosion) and don't compute precisely.

Problem 8A.12 — Craps (compound conditioning)

In craps you roll two dice: win immediately on 7 or 11; lose on 2, 3, 12; otherwise your roll becomes "the point" and you win if you re-roll the point before rolling a 7. P(win)?

Full solution. Immediate win: 8/36. For each point p with probability w(p) = P(p before 7) = (#ways p)/(#ways p + 6) [race template, 8A.3]: - Point 4 or 10 (3 ways each): w = 3/9 = 1/3 → contribution 2 × (3/36)(1/3) = 2/36 - Point 5 or 9 (4 ways): w = 4/10 → 2 × (4/36)(0.4) = 3.2/36 - Point 6 or 8 (5 ways): w = 5/11 → 2 × (5/36)(5/11) ≈ 4.55/36 Total ≈ (8 + 2 + 3.2 + 4.55)/36 = 17.75/36 ≈ **0.493.** Slightly worse than a coin flip — exactly the kind of "is this bet fair?" judgment the test wants. The fast version: know craps ≈ 49.3% and, more importantly, the decomposition pattern: immediate cases + Σ P(point) × P(race win).

Problem 8A.13 — Two uniform points

X and Y are independent uniform on [0,1]. (a) P(|X−Y| < ½)? (b) E[|X−Y|]?

Full solution. (a) Picture the unit square; the band |x−y| < ½ excludes two corner triangles each with legs ½: excluded area = 2 × ½ × (½)² = ¼. **P = 3/4.** (b) Known result: E|X−Y| = **1/3**. (Generally for [0, L]: L/3.) **Template:** any two-uniform-variables question → draw the unit square, find the area. Bands, triangles, and quarter-circles cover virtually all cases.

Problem 8A.14 — Secretary problem (know the headline)

You interview 100 candidates in random order, must accept/reject each on the spot, want the single best. Optimal strategy and success probability?

Solution sketch. Observe the first n/e ≈ 37 candidates without choosing; then accept the first candidate better than all you've seen. Success probability ≈ **1/e ≈ 0.37.** Tests ask this as "what fraction should you observe first?" (≈37%) or "chance of getting the best?" (≈0.37). Both answers are 1/e — a happy coincidence worth remembering.

Problem 8A.15 — Conditional expectation with re-pricing

A box has 2 fair coins and 1 two-headed coin. You pick one at random and flip it twice: both heads. You'll now flip it a third time. P(third flip is heads)?

Full solution. Posterior after HH: fair → (2/3)(1/4) = 1/6 weight; trick → (1/3)(1) = 1/3 weight. Normalize: P(trick) = (1/3)/(1/3 + 1/6) = 2/3; P(fair) = 1/3. P(H next) = (2/3)(1) + (1/3)(½) = **5/6 ≈ 0.83. Two-stage template:** update beliefs with Bayes first, then compute the forward probability under the posterior. Never mix the stages.

Problem 8A.16 — The envelope of "at least one" traps (boss level)

Roll 6 dice. Which is bigger: P(at least one 6 in 6 rolls), P(at least two 6s in 12 rolls), or P(at least three 6s in 18 rolls)? (Newton–Pepys problem)

Full solution. Each scenario has expected sixes = 1, 2, 3 respectively — equal in "expectation per requirement," so intuition says they're equal. Compute: - $P(\geq 1 \text{ in } 6) = 1 - (5/6)^6 = 1 - 0.335 =$ **0.665** - $P(\geq 2 \text{ in } 12) = 1 - (5/6)^{12} - 12(1/6)(5/6)^{11} = 1 - 0.112 - 0.269 =$ **0.619** - $P(\geq 3 \text{ in } 18) \approx$ **0.597** The first is largest — distributions spread out, and needing *more* successes (even proportionally) is harder. **Exam takeaway:** when comparing "k successes in kn trials" families, fewer-required wins. Also note the reusable computation: $P(\geq 2) = 1 - P(0) - P(1)$, with $P(1) = n \cdot p \cdot (1-p)^{n-1}$.

8B. NumberLogic — Hard set (12 sequences)

H-1. 2, 7, 22, 67, 202, ? *Solution:* ratios ≈ 3 → test ×3+1: 2→7 ✓, 7→22 ✓, 22→67 ✓, 67→202 ✓ → 202×3+1 = **607**.

H-2. 1, 2, 5, 14, 41, 122, ? *Solution:* ×3−1: 1→2 ✓, 2→5 ✓, 5→14 ✓ → 122×3−1 = **365**. (Spot "≈×3" from 41→122 first, then fit the correction.)

H-3. 2, 5, 26, 677, ? *Solution:* each term = $\text{previous}^2 + 1$: $2^2+1=5$, $5^2+1=26$, $26^2+1=677$ → 677^2+1 = **458,330**. Recognize explosive growth → squaring, not multiplying.

H-4. 3, 5, 10, 12, 24, 26, 52, ? *Solution:* alternating +2, ×2: 3+2=5, 5×2=10, 10+2=12, ×2=24, +2=26, ×2=52 → +2 = **54**.

H-5. 12, 15, 21, 24, 30, 33, ? *Solution:* gaps +3, +6, +3, +6, +3 → next gap +6 → **39**. (Alternative reading: each term + its digit sum: 12+3=15, 15+6=21, 21+3=24, 24+6=30, 30+3=33, 33+6=39 — same answer, which is why the test likes it.)

H-6. 4, 6, 10, 14, 22, 26, ? *Solution:* halve everything: 2, 3, 5, 7, 11, 13 → primes ×2 → next prime 17 → **34**. **Trick: when a sequence is all even, factor out 2 and look again.**

H-7. 1, 10, 2, 20, 200, 4, 40, 400, 4000, ? *Solution:* blocks growing in length: (1,10), (2,20,200), (4,40,400,4000) — block starters 1, 2, 4 (doubling), each block multiplies by 10. Next block starts at **8**.

H-8. 1/2, 2/3, 3/5, 5/8, ? *Solution:* numerators 1,2,3,5 and denominators 2,3,5,8 — both Fibonacci, denominator = next term after numerator → **8/13**.

H-9. 0, 1, 8, 27, 64, ? *Solution:* $(n-1)^3$ → **125**. Easy once seen — the 0 start is the disguise. A leading 0 usually signals "shifted famous sequence."

H-10. 5, 7, 12, 19, 31, 50, ? *Solution:* each term = sum of previous two: 5+7=12, 7+12=19, 12+19=31, 19+31=50 → 31+50 = **81**. Fibonacci recursion with non-standard seeds — always test $a_{n-1}+a_{n-2}$ when differences reproduce the sequence itself.

H-11. 95, 115.2, 138.24, ? *Solution:* 115.2/95 = 1.2128… not clean; try 115.2 = 96×1.2 — recheck: 95×1.2 = 114 ✗. Try +20.2, +23.04: ratio of gaps = 23.04/20.2 ≈ 1.14 ✗. Try ×1.2 on 96: the sequence is actually 138.24/115.2 = 1.2 exactly ✓ and 115.2/95 = 1.2126 ✗ → first ratio breaks. Decimals like .24 scream ×1.2 twice (1.44); test 95→? : 95 × 1.44 = 136.8 ✗. Resolution: gaps 20.2 and 23.04 — second differences? 2.84. Nothing clean → **flag and skip; revisit only with spare time.** *This problem is included deliberately: recognizing an ill-posed/too-slow question and skipping IS the skill.* (If forced: extend gap ratio 23.04/20.2 ≈ 1.1406 → next gap ≈ 26.28 → ≈ 164.5.)

H-12. 6, 24, 60, 120, 210, 336, ? *Solution:* $n(n+1)(n+2)$: 1·2·3=6 … 7·8·9 = **504**. Confirm via differences: 18, 36, 60, 90, 126 → second differences 18, 24, 30, 36 (+6) → next first-difference 126+42=168 → 336+168 = 504 ✓. **Two independent routes agreeing = answer locked.**

8C. Likelihood-list — Hard set (3 problems)

Problem 8C.1 — Near-tie resolution

Roll 4 dice. Rank: (a) at least one 6, (b) exactly one 6, (c) sum ≥ 20, (d) all faces different.

Full solution. (a) $1 - (5/6)^4 = 1 - 625/1296 \approx 0.518$ (b) $4 \times (1/6)(5/6)^3 = 500/1296 \approx 0.386$ (d) $(6 \cdot 5 \cdot 4 \cdot 3)/6^4 = 360/1296 \approx 0.278$ (c) Sum ≥ 20 means averaging 5 per die. Count via complement symmetry: sum of 4 dice is symmetric around 14; P(sum ≥ 20) = P(sum ≤ 8). Quick count of sums ≤ 8 with 4 dice: 4→1, 5→4, 6→10, 7→20, 8→35 ways = 70/1296 ≈ 0.054. **Rank: a > b > d > c.** The (b) vs (d) gap (0.39 vs 0.28) is the discriminating comparison; the symmetry trick for (c) avoids the only genuinely slow computation.

Problem 8C.2 — Conditional ranking

A card is drawn and you're told it's a face card (J/Q/K). Rank: (a) it's a heart, (b) it's a king, (c) it's red, (d) it's the king of hearts.

Full solution. Restricted universe = 12 face cards. (a) 3/12 = 0.25 · (b) 4/12 ≈ 0.33 · (c) 6/12 = 0.5 · (d) 1/12 ≈ 0.083 **Rank: c > b > a > d.** Trap: unconditionally, hearts (13/52) beat kings (4/52) — conditioning on "face card" reverses (a) and (b) because all kings survive the conditioning but only 3 hearts do. **Conditioning changes rankings; recount inside the restricted universe, never carry over unconditional intuitions.**

Problem 8C.3 — Mixed structure + base rates

A random Singapore resident is selected. Rank: (a) owns a car, (b) took the MRT this week, (c) owns a car AND took the MRT this week, (d) has a driver's licence, (e) owns a car but did NOT take the MRT this week.

Full solution. Structure first: (c) ≤ (a) and (e) ≤ (a), and (c) + (e) = (a) exactly (partition of car owners). So (a) strictly exceeds both, and (c) vs (e) is the only judgment call. Base rates: Singapore's vehicle quota keeps car ownership low (~1/3 of households, lower per resident, say ~15–20%); licences are far more common (~50%); MRT weekly usage is very high (~70%+). (b) ≈ 0.7 >

(d) ≈ 0.5 > (a) ≈ 0.18. For (c) vs (e): even car owners in Singapore commonly ride MRT weekly → (c) > (e), each below 0.18. **Rank: b > d > a > c > e.** Note how little precision was needed: the partition identity did half the work, coarse base rates did the rest.

8D. Intervals — Hard Fermi set (4 problems, full decompositions)

Problem 8D.1 — Words spoken in a lifetime

- Words/day: people speak ~2–3 hours of cumulative talk; speech ≈ 130 wpm → 150 min × 130 ≈ 16,000/day (use 15k)
- Speaking years: ~75 → days ≈ 27,000
- Total ≈ 15,000 × 27,000 ≈ **4×10^8 words. Interval: [1×10^8, 1.5×10^9]** (k ≈ 3–4: daily talk time is the shaky factor).

Problem 8D.2 — Total mass of all living humans

- 8×10^9 people × average mass ~50 kg (children pull the worldwide average well below adult 62–70 kg)
- ≈ **4×10^{11} kg = 400 million tonnes. Interval: [3×10^{11}, 6×10^{11}] kg** — tight (k ≈ 1.3) because both factors are well known. **Lesson: interval width must track factor certainty, not a fixed habit.**

Problem 8D.3 — Passengers through Changi Airport per day

- Pre-pandemic-recovered Changi handles ~60M passengers/year (anchor: it's a top-20 global hub; top hubs run 60–100M/yr)
- 60M/365 ≈ **~165,000/day. Interval: [100k, 250k].**
- No anchor for Changi? Rebuild: ~360 destinations × ~? Simpler: ~1,000 flights/day × ~150 pax average ≈ 150k ✓ — two routes agreeing lets you *tighten* the interval; that's when narrow is justified.

Problem 8D.4 — Daily global oil consumption

- Anchor either the total (~100M barrels/day — worth memorizing) or rebuild: 1.5B vehicles × ~4 L/day ≈ 6×10^9 L ≈ 38M barrels (159 L/barrel) for road transport; transport ≈ 50–60% of oil use → total ≈ 70M; industry/heating tops it up
- **Estimate ~100M barrels/day. Interval: [60M, 150M].**

8E. Orderbooks — Hard set (4 problems)

Problem 8E.1 — Multi-bundle netting (vector method)

Instrument	Contents	Bid	Ask
AB	1A + 1B	14	15
BC	1B + 1C	9	10
ABC	1A + 1B + 1C	24	25
B	1B	2	3

Full solution. Write item vectors and hunt for a zero-sum combination: Buy AB (+1A,+1B, −15 cash) + Buy BC (+1B,+1C, −10) = holding A + 2B + C, cash −25. Sell ABC (−1A,−1B,−1C, +24) → left with +1B, cash −1. Sell B at bid 2 → flat, cash **+1 ✓.** Check the reverse for completeness: Buy ABC 25 + Buy B 3 = 28; Sell AB 14 + Sell BC 9 = 23 → −5 ✗. **Trade: buy AB, buy BC, sell ABC, sell B → riskless +1.** The discipline: track *items* and *cash* as separate columns; an arbitrage is items = 0, cash > 0.

Problem 8E.2 — Sizing against displayed volume

Widget: venue 1 ask 99 × 50 units available; venue 2 bid 101 × 30 units. Also venue 1 bid 98 × 200, venue 2 ask 102 × 200. Max riskless profit?

Full solution. The only cross is buy@99 / sell@101 (+2/unit). Size = min(50, 30) = 30 units → **profit 60.** Trap check: after 30 units, remaining quotes are bid 98 / ask 99 → no cross, stop. Greedy candidates who answer "50 units" or chase the 98/102 pair fail this. **Size on the smallest leg; re-examine the book after the cross is consumed.**

Problem 8E.3 — Triangle with executable directions

Conversions available: A→B at 2.0 (each A yields 2 B); B→C at 3.0; C→A at 0.18 (each C yields 0.18 A). Reverse legs not offered. Arbitrage?

Full solution. Only one loop direction exists: A→B→C→A. Multiply yields: 2.0 × 3.0 × 0.18 = 1.08 > 1 → **yes: each loop turns 1 A into 1.08 A, +8% per cycle.** Had the product been 0.93, the answer would be "no trade" — you cannot run the loop backward unless reverse quotes exist. **Always check which directions are actually executable before inverting.**

Problem 8E.4 — The mid-price mirage (no-arb recognition)

Instrument	Bid	Ask
X	9	11
Y	4	6
XY bundle	16	18

Full solution. Mid-prices: X = 10, Y = 5, bundle = 17 → bundle looks 2 “rich” vs parts (15). But execute: sell bundle at bid 16, buy parts at asks 11 + 6 = 17 → **−1 ✗.** Other direction: buy bundle 18, sell parts 9 + 4 = 13 → −5 ✗. **No arbitrage. The spread eats the mispricing.** This is the most realistic trap in the whole test: mids suggest an edge that crossing the spread destroys. Profit is computed bid-vs-ask, never mid-vs-mid.

End of guide. Train timed, stay +EV, and remember: a skipped −EV question is a point earned.