The Putnam Circuit.

Necessary and sufficient reaction circuits, mapped from the TMUA9 WHAT→HOW→WHY protocol onto the Putnam corpus (1985–2025 A1–B6).

0. Translation of the TMUA9 method to Putnam

TMUA command terms are surface wording nudges in a multiple-choice stem.

Putnam has no MCQ scaffolding, so the "command term" layer becomes **structural cues**: the object class the problem hands you (integer sequence, polynomial, matrix, function, sum, inequality, finite set, probability space) plus the **quantifier shape** of the ask (prove for all / find all / determine the value / show existence / compute mod p / find the minimum). The reaction circuit is the reflex move that the structural cue should trigger before you write a single line of computation for the question.

The mission statement is still unchanged: build reaction circuits so I react a certain way when I read the problem's structural cues.

Analysis unit = one problem, not one paper. Do the WHAT → HOW → WHY against the official solutions (Kedlaya/Poonen/Vakil compilations, AoPS threads). Do not grind an unseen problem before this bank is internalized on problems you've already seen solved; same rule TMUA9 gives: past-paper analysis first, cold-solving later on is the performance phase, not the skill-building phase.

Right now, coming from a personal diagnostic, my database is simply too small; the sum of pascal triangles, the Hockey stick theorem, these things are simply not intuitive for me right now. Until I am able to create my own questions, I need to work my way up the ladder of the Knowledge Taxonomy of Bloom.(Knowledge: Recalling facts, definitions, or basic concepts. Comprehension: Grasping the meaning of information. Application: Using rules or knowledge in new situations. Analysis: Breaking information down into component parts. Synthesis: Combining parts to form a new, coherent whole. Evaluation: Making value judgments based on criteria)
For me, I simply do not know all the content, that’s the first markovow that I must get done; to read all the contents in the

blueprint first and apply them.

Layer 1 WHAT: juxtapose your attempted approach vs the official one, verbatim.

Layer 2 HOW: find the exact line of bifurcation, extract the one sentence you had to think to follow the official path.

Layer 3 WHY: name the axiom/theorem that makes that sentence forced, not optional.

Layer +α backtest: what was the first cue, what circuit should have fired, where did you hesitate, what's the reusable rule.

---

1. Algebra, Polynomials, Functional Equations

Cue: "polynomial with integer/rational coefficients"
Reaction: immediately test rational root theorem, reduce mod small primes (2, 3), and check P(a) − P(b) divisible by a − b.

Discernment: if the question asks about roots or factorization → RRT + mod-p reduction; if it asks about values at integers → the divisibility identity a−b | P(a)−P(b).

Why: $\mathbb{Z}[x]$ is closed under evaluation homomorphisms mod p; the divisibility identity follows directly from the factor theorem applied to P(x)−P(y).

Adjacent: Eisenstein's criterion, reduction mod p to prove irreducibility, Gauss's lemma.

Drill: Putnam 2001 B1 (rational root under substitution), 1994 A5 (mod-p polynomial argument), 1999 A6, 2007 A5, 2013 B1.

Cue: functional equation with f(x+y) or f(xy) structure
Reaction: test f = constant, f = identity, f = 0 first to bound the answer space, then set specific substitutions (x=y, y=0, y=1, x=-x) before attempting generality.

Discernment: additive shape → Cauchy-type, look for $\mathbb{Q}$-linearity then continuity/monotonicity to force $\mathbb{R}$-linearity; multiplicative shape → take logs or test power functions.

Why: Cauchy's functional equation is only pinned to linear solutions once you add a regularity condition (continuity, monotonicity, boundedness); without it, pathological solutions exist via Hamel bases — so the circuit is "find where the problem smuggled in the regularity condition."

Adjacent: fixed point substitutions (set x=y to find f(f(x))), induction to extend from $\mathbb{Z}$ to $\mathbb{Q}$ to $\mathbb{R}$.

Drill: 2007 B2, 2011 A6, 1996 A5, 1988 B6, 2015 A2.

Cue: symmetric expression in multiple variables (sums, products, cyclic sums)

Reaction: reroute to elementary symmetric polynomials / Newton's identities before expanding by brute force.

Discernment: if the answer is symmetric under permutation of variables → symmetric function machinery; if it's cyclic but not fully symmetric → look for SOS (sum of squares) or Schur-like structure instead.

Why: the ring of symmetric polynomials is generated by elementary symmetric polynomials (fundamental theorem of symmetric functions) — any symmetric target is expressible in that basis, which is almost always shorter than direct expansion.

Adjacent: Vieta's formulas, power sum identities, AM-GM applied to the elementary symmetric terms.

Drill: 1971 A3-style, 2004 A2, 2010 A5.

Cue: inequality to prove, no equality case given

Reaction: identify which of {AM-GM, Cauchy-Schwarz, rearrangement, convexity/Jensen, SOS} matches the exponent/structure signature before touching calculus.

Discernment: products vs sums of the same terms → AM-GM; sums of products (dot-product shape) → Cauchy-Schwarz; ordered sequences paired → rearrangement; single-variable convex function of a weighted average → Jensen; quadratic-in-disguise → SOS.

Why: each inequality is the same convexity fact (Jensen) specialized to a particular kernel — recognizing which specialization avoids reproving convexity from scratch with derivatives.

Adjacent: power mean inequality, tangent line trick, smoothing/SOS substitution.

Drill: 1998 A6, 2003 A6, 2016 A5, 2020 A2.

---

2. Number Theory

Cue: "for all primes p" or "mod p" appears explicitly Reaction: reduce the whole statement mod p immediately, and check Fermat's Little Theorem / Wilson's theorem applicability before any other manipulation. Discernment: statement about (p−1)! or products over a residue system → Wilson; statement about a^(p−1) or a^p → Fermat/Euler; statement about quadratic residues → Legendre symbol and quadratic reciprocity. Why: (ℤ/pℤ)* is a cyclic group of order p−1 — FLT is just Lagrange's theorem in disguise; the moment you see mod p, you have a finite cyclic group to exploit, not an open-ended integer problem. Adjacent: lifting the exponent (LTE) for prime power divisibility, order of an element mod p. Drill: 1996 B4, 2005 B4, 2014 B3, 1991 B5.

Cue: infinite descent flavor — "no positive integer solutions" or minimal counterexample language Reaction: assume a minimal counterexample exists (well-ordering) and construct a strictly smaller one; don't attempt direct proof first. Discernment: if the equation is homogeneous or scalable (like Fermat-type a^n+b^n=c^n or Pell-like), descent is almost always the shorter route than case analysis. Why: the well-ordering principle on ℕ guarantees a minimal element exists in any nonempty subset — descent converts "prove none exist" into "derive a contradiction from the smallest one," which is strictly easier than handling the general case. Adjacent: Vieta jumping (descent via root-swapping in a quadratic), parity/mod 4 obstruction as the descent step. Drill: 1995 A6 (Vieta jumping), IMO 1988 P6 analog problems, 2000 A5.

Cue: sum or count of divisors, or "number of ways to write n as..." Reaction: translate to a generating function or Dirichlet-convolution identity before attempting direct combinatorial counting. Discernment: multiplicative structure across n → generating functions

/ Euler product; additive partition-type counting → recursive generating function in x. Why: divisor-sum functions are multiplicative, so their generating Dirichlet series factor over primes — this converts a hard combinatorial sum into a product you can bound or compute termwise. Adjacent: Möbius inversion, floor-function sum manipulation (Hermite's identity), Legendre's formula for p-adic valuation of n!. Drill: 2018 A2, 1997 B3, 2009 A5.

Cue: floor/ceiling function or "greatest integer" appears Reaction: split into integer part + fractional part algebraically; look immediately for a telescoping sum in disguise. Discernment: single floor term summed over a range → Hermite's identity or direct telescoping; floor inside an equation to solve → bound between consecutive integers and squeeze. Why: $\lfloor x \rfloor$ is a step function whose only usable algebraic property is $x - 1 < \lfloor x \rfloor \leq x$ — every floor-function circuit reduces to converting that inequality into a squeeze, because there is no other lever available. Adjacent: Beatty sequences, Legendre's formula, three-distance theorem. Drill: 2000 A2, 2012 A4.

---

3. Combinatorics

Cue: "find the number of ways" with a recursive-looking constraint Reaction: set up the recurrence in terms of smaller n before searching for a closed form; try n=1,2,3 by hand to find the pattern only after the recurrence is written symbolically. Discernment: constraint depends only on the immediately preceding element(s) → linear recurrence, solve via characteristic equation; constraint is global (all pairs, all subsets) → generating function instead. Why: any counting problem with a "local" dependency structure satisfies the Markov property on the count sequence, which guarantees a linear recurrence exists — hunting for a closed form before finding the recurrence throws away this guarantee and forces guesswork. Adjacent: transfer matrix method, Catalan-number recognition (balanced-parenthesis shape), bijective proof search. Drill: 2006 A4, 2010 A3, 1990 B4.

Cue: "at least" / "there exist" existence claim on a finite structure Reaction: reach for pigeonhole first, then extremal principle (take the largest/smallest object with a property and derive a contradiction if it isn't already what you want), before probabilistic method. Discernment: fixed small bound with many objects → pigeonhole; a single "extreme" object must satisfy the

property → extremal principle; bound only holds "on average" or asymptotically → probabilistic/counting argument (expected value ≥ target implies some instance ≥ target). Why: pigeonhole is just a counting inequality (n objects, k boxes, k<n forces a box with ≥2); the extremal principle is well-ordering again; the probabilistic method is Markov's inequality run backwards — all three are the same tool (an averaging/counting inequality) applied at different levels of aggregation, so the cue "there exists" should trigger checking averages before attempting direct construction. Adjacent: double counting (Fubini for sums), Hall's marriage theorem, graph coloring extremal arguments. Drill: 2003 A1, 2014 A2, 1988 A4, 2001 B4.

Cue: probability problem with expectation being asked ("expected number/value") Reaction: decompose into a sum of indicator random variables and use linearity of expectation before attempting to compute the full distribution. Discernment: "expected value of a count" → linearity of indicators, almost always; "probability of a specific event" (not a count) → direct combinatorial ratio or recursive conditioning. Why: linearity of expectation holds unconditionally, even for dependent random variables — this is the single most underused fact in Putnam combinatorics, because it lets you avoid ever computing the joint distribution. Adjacent: symmetry arguments (exchangeability to equate expectations across positions), conditioning/tower property, variance via $E[X^2]-E[X]^2$. Drill: 2005 A6, 2013 A6, 1994 B3.

---

4. Real Analysis, Calculus, Sequences/Series

Cue: "prove the limit exists" or "prove the sequence converges" Reaction: reach for monotone + bounded first; only if that fails, check Cauchy criterion or contraction mapping. Discernment: sequence defined by a recursive average or convex combination → monotone-bounded almost always works; sequence not monotone but self-correcting → contraction mapping (show $|a_{n+1}-L| \le r|a_n-L|$, $r<1$). Why: the monotone convergence theorem is the cheapest existence proof available in $\mathbb{R}$ because it needs no explicit limit value — it converts "does it converge" into "is it eventually monotone and bounded," which is usually a one-line induction, whereas Cauchy or contraction require estimating rates. Adjacent: Stolz-Cesàro (discrete L'Hopital), telescoping to find the actual limit once existence is known, squeeze theorem. Drill: 1990 A4, 2004 B1, 2020 A5.

Cue: differentiable/continuous function on [a,b] with a value condition (f(a)=f(b), or an integral condition) Reaction: reach for Rolle's theorem / MVT immediately if two point-values are equal or an integral is zero; reach for IVT if you need an existence-of-root claim. Discernment: equal endpoint values or a derivative condition to produce → Rolle/MVT chain (possibly applied twice, to f and f′); integral equals a specific value and you need a pointwise conclusion → mean value theorem for integrals. Why: Rolle's theorem is the single hypothesis that converts "the function returns to the same value" into "somewhere the instantaneous rate of change is exactly zero" — it's the only tool in single-variable analysis that produces a derivative-value guarantee for free, so any problem handing you equal values at two points is handing you this theorem. Adjacent: Taylor's theorem with Lagrange remainder for higher-order versions, Cauchy's mean value theorem for ratios of functions. Drill: 1993 B4, 2002 A6, 2011 A5, 1998 B1.

Cue: series to evaluate, with a factorial, binomial coefficient, or power in the general term Reaction: identify it as a known generating function evaluated at a point (e^x, 1/(1-x), (1+x)^n, ln(1+x)) before attempting term-by-term manipulation. Discernment: factorials in denominator → e^x family; binomial coefficients → (1+x)^n family, often needs differentiation or integration of the generating function to shift the power; alternating signs with 1/n → ln(2) family. Why: every "nice" convergent series on the Putnam is secretly a generating function evaluated at a specific x, because that's the only systematic way problem-setters guarantee a closed form exists — matching the term structure to the right generating function converts an open-ended summation into an algebra problem. Adjacent: differentiating/integrating a known series termwise to shift indices, Abel summation, generating function manipulation (substitution, multiplication). Drill: 2001 A4, 1992 A5, 2019 A4.

---

5. Linear Algebra & Matrices

Cue: "matrix A such that A commutes with..." or trace/determinant condition given Reaction: reach for eigenvalue/eigenvector language immediately, and check whether the condition forces diagonalizability before doing any entrywise computation. Discernment: condition on trace and determinant only (not full matrix) → work with eigenvalues symbolically (they determine trace and det for any size via elementary symmetric functions of eigenvalues); condition on commuting with a specific known matrix → the space of matrices

commuting with a diagonalizable matrix is exactly the matrices diagonal in the same eigenbasis. Why: trace and determinant are the first and last elementary symmetric functions of the eigenvalues — any problem stated purely in trace/det is really a problem about the eigenvalue multiset, and working there is always lower-dimensional than the entrywise matrix. Adjacent: Cayley-Hamilton to reduce high powers of A to a linear combination of I and A, Jordan form for non-diagonalizable edge cases. Drill: 2016 B4, 2008 A5, 1997 A3.

Cue: "prove the determinant is nonzero" / "prove a system has a unique solution" Reaction: reach for linear independence of rows/columns first; check if it reduces to showing a certain vector isn't in the span of the others, rather than expanding the determinant directly. Discernment: structured matrix (Vandermonde-like, circulant, tridiagonal) → recognize the closed-form determinant formula instead of cofactor expansion. Why: determinant ≠ 0 is definitionally equivalent to linear independence of the columns — reformulating in terms of independence usually reveals a one-line argument (e.g. distinct evaluation points → Vandermonde nonzero) that cofactor expansion buries in algebra. Adjacent: Vandermonde determinant formula, Cauchy-Binet, Schur complement for block matrices. Drill: 1999 B2, 2007 B4.

---

6. Geometry

Cue: circle(s) with a single point of tangency or intersection condition Reaction: draw the auxiliary line connecting the center(s); reach for power of a point before coordinate bashing. Discernment: two circles tangent → line through both centers passes through the tangency point; a point with equal tangent lengths to a circle → power of a point directly gives the algebraic relation. Why: a circle is defined as the set of points equidistant from its center — any tangency or intersection statement is a distance statement, and the center is definitionally the only point from which every relevant distance is controlled, so the auxiliary line to the center is forced by the definition, not a trick. Adjacent: radical axis for two circles, homothety centered at the tangency point. Drill: matches the TMUA9 circle circuit directly — same underlying fact, applied at Putnam depth in 2003 B6-style configuration problems.
Adjacent-adjacent: for triangle configurations, default to vectors or complex numbers (unit circle circumcircle) before pure synthetic angle-chasing, since Putnam geometry is rare enough that a

computational fallback should fire immediately if no synthetic circuit fires within two minutes.

---

7. Meta-circuits (fire before any domain-specific one)
Cue: "find all" (functions / integers / triples satisfying a condition) Reaction: conjecture the answer set FIRST from small cases (n=1,2,3, or simplest function forms), then split the proof into (a) verify the conjectured set works, (b) prove nothing else works. Never attempt (b) before (a) is written down. Why: "find all" is a two-sided claim disguised as one instruction; treating it as one-sided is the single most common source of wasted time on this problem type, because you end up proving uniqueness for a set you haven't even fixed yet.

Cue: problem gives an inequality with a specific numeric constant (not a general n) Reaction: check if the constant is tight (equality case exists) before choosing a technique; a tight constant means the technique must produce equality somewhere, which usually pins down which inequality (AM-GM vs Cauchy vs Jensen) is intended. Why: problem-setters choose the sharpest available inequality for the stated constant — the constant itself is a command term pointing at the equality case, which is the fastest way to reverse-engineer the intended tool.

Cue: you are stuck for more than 3-4 minutes with no progress
Reaction: switch representation. (Go all in on the antithesis)
• Generalised form <-> Specific examples
• Algebraic <-> try a small-case table
• Synthetic geometry → try coordinates/complex numbers/vectors.
• Combinatorial → try a generating function or a bijection to a known structure.

Never ruminate the same representation past this mark, you must keep the ship rolling at least in the opposite direction.

Why: being stuck is okay, but being trapped in a cycle of redundant rumination is not ideal. When you are stuck, it is better to specify what it is that you are stuck with, and try the antithesis or attempt resolving that one issue rather than repeating what has already been proven to be right.

Having slower velocity of processing alongside the dance to the solution, is not evidence you need to compute harder. Which matches the TMUA9 principle that speed comes from rerouting, never from working faster inside a bad path.

Backtesting checklist (run after every problem, Putnam-adapted)

1. What was the first structural cue (object type + quantifier shape) that caught our eye?
2. Which circuit should have fired? Did it run, and if not, why not?
3. Where exactly did you hesitate or default to brute force ("zombie maths")?
4. What was the actual bifurcation point vs. the official solution?
5. What is the one-sentence critical thought you had to self-generate?
6. What is the underlying axiom/theorem that makes that thought forced, not optional?
7. Extract the reusable general rules, and add it to this bank if it isn't already covered above.

Cadence: one Putnam problem through full WHAT→HOW→WHY per session, analysis time 2–3x the solve/read time, never end a session on attempting a new problem — always end on extraction.