The Blueprint for Oxford/Cambridge Interviews.

Jamie Baek · TMUA9

The North Star, the trinity of skills reverse engineered from it, the attitude, the generalisation machinery, the cadence, and the interview day protocol.

Part of the print series: Blueprint, Redprint, Greenprint. For Y12, Y13 and gap year applicants to STEM and Economics at Oxford and Cambridge.

Free, always. tmua9.com

9.

The Blueprint.

Jamie Baek · TMUA9

This document is a comprehensive guide to Oxbridge interviews. Specifically, it is for Y12, Y13 and gap year students applying to STEM or Economics courses at Oxford or Cambridge.

It is the sequel to the print series for admissions tests (the Blueprint, the Redprint and the Greenprint for the TMUA, TARA and ESAT). So let us start by seeing how the next step, the interview, differs from admissions test preparation.

There are overlapping skills, intuition and pattern recognition above all. But the strategy you must take for interview preparation is going to be completely different from how you prepared for the admissions test, because the skillsets that make you excel in each process diverge too.

In your interview, there are two fellows sitting in a room with you for about 30 minutes, and the only output they can see is what you write and what you say. Everything else that goes on inside your head is invisible to them, thus cannot be assessed. That single fact is the reason why how you approach interviews must never be how you did your preparation for admissions tests.

In the admissions test, your ability to vocalise your intuition and your ability to express a chain of reasoning as a coherent proof are not assessed at all. The test needs one thing: intuition. The interview is not like that. Whether you can solve every problem has surprisingly low bearing on how your performance is graded (Mathematics at Trinity College, Cambridge being a rare exception). How transparently you express the way you work through a problem is what is vital.

To achieve outputs, there are certain skills and characteristics we must cultivate to maximise the probability of the desired outcome. You will already know most of the knowledge presented here. What is aimed to be done is to present a frame for thinking about it, so that you can calibrate how to prepare deliberately in the confined time we have remaining.

This document is split into three sections.

1. **The North Star.** What ideal performance looks like, and the three core skills reverse engineered from it: Vocalisation, Intuition and Metacognition. Presented as a mind map alongside a taxonomy, so that you know exactly which skillset you are training at any given moment.
2. **Attitude.** The ways to think of the interview so that you walk in as a collaborator, not as a defendant.
3. **The actual blueprint.** The mission statement, the generalisation machinery with worked examples, the questions to ask back to the interviewer, the

cadence of preparation, and the interview day protocol. It is not recommended to skip to this part; it is an amalgamation of the prior sectors.

Before we start, I want to make the same thing clear that I did in the Blueprint. The guidelines here will also be extremely unorthodox against the contemporary advice out there:

> **Mainstream advice is not meant to provide a guideline that yields the anomalous results we are aiming for. Normal advice brings normal actions, thus derives normal results that are merely mediocre.**

Well known, vague, mainstream advice for interviews goes something like:

- "Just be yourself and you will do fine!"
- "Read your personal statement and read around your subject."
- "Practise talking about maths with your teacher."
- "They just want to see how you think."
- "Just do more maths!"

These are merely vacuous platitudes. Everyone has heard these empty statements, but none of this "advice" directs us towards specific action steps that contribute to the outcomes we want. Nobody tells you what "how you think" consists of. This is why so many students end up doing low return-on-investment work within the confined time they have, with zero improvement in the skills that actually matter: rote repeating random questions they come across, with no clear aim, hoping their thinking is good enough for the interview. Hope is neither a method nor a strategy. We will decompose "how you think" into trainable parts, and train each part deliberately.

Section 1: The North Star.

There are very specific skillsets that Oxbridge admissions tutors want to see in you, and there is a very deliberate way to hone them. First, what does the manifestation of an ideal Oxbridge interview performance look like? What is our North Star?

1. Being able to cohesively vocalise your journey from reading the question to getting to the answer, transparently.
2. Knocking out complex and abstract problems with precision, consecutively, in short periods of time.
3. The curiosity to transcend the presented question into generalised principles, and to explore alternative, even contrary, approaches.

The dichotomy comes from the nature of pure mathematical problem solving. Almost all revelations are non-verbal, capricious and messy. The interviewers know this too, but it is your responsibility to communicate what you are thinking right now. That is why guidance is offered by the interviewers: to trigger those revelations out of you.

Especially for STEM, you are competing against a large pool of people who have done olympiad training for nearly a decade. Yet there are surprisingly many cases where these great problem solvers end up with poor interview performance. Why? The revelation happens in an implicit part of you that does not speak, but the interview only rewards what is explicitly expressed. Usually, how olympiad problems get solved is that the student sits in silence for a while, then says "I think it is 12", and the interviewer has learned nothing about them, thus cannot assess them. Which is why they end up saying things like:

- "This and this and this triangle has to be the same."
- "It is quite obvious if you think about it."
- "This dot here is on this line."

instead of expressing it in technical terms. Also, most intuition comes from speculative guesses ("I think this might be true"), then logic dilates later to catch up to it, so you should also be able to balance the order in which you present these.

The logical conclusion is that the entire preparation is the construction of a bridge between a non-verbal engine and a verbal output, while keeping the engine (intuition) running at full velocity in parallel. Great problem solvers have trained the engine. Building the bridge is a completely different skillset, and it is not normally taught, because it is not relevant to mathematical intuition itself.

There are three core skills we will focus on.

1. Vocalisation.

- The ability to translate non-verbal thoughts into verbal expressions, including writing.
- The ability to vocalise those converted expressions synchronously, in parallel with solving the problem.

You can write more than you can speak, and you know more than you can write.

Speaking ⊂ Writing ⊂ Knowledge

Converting everything you "know" into verbal expression (writing or speaking) is a contractionary process, and speaking out what you can write is an even more contractionary process. So the interview tests you on the innermost, most compressed set. If you only ever practise on the outermost set (solving in your head, pondering, quantum leaping between critical thoughts) you are training the one thing they cannot observe.

The two bullet points are two different skills and they fail in two different ways. **Translation** fails when you have an idea and cannot find the words for it, so you say "um, so, like, you kind of..." and the idea dies in your mouth. **Synchrony** fails when you can describe your thinking perfectly, but only after the fact: you go silent, solve it, then narrate. The interviewer does not want a post-match report. They want the live commentary. Synchrony is the harder one and it is the one to drill.

2. Intuition.

Having higher calibre speculations, thus decreasing the margin of error on your guesses about how to solve a problem. Intuition is a byproduct of two factors, and you need to optimise both equally:

- Scale of the database.
- Coherence within that database.

I want to kill the word "talent" here, the same way I killed it in the Blueprint. Intuition feels like magic from the inside because the database lookup is instant and silent. But it is a lookup. When a Trinity fellow "just sees" that a sum telescopes, it is because they have seen two thousand sums and the ones that telescope are all filed together under one principle. Scale is the two thousand. Coherence is the filing. A student with scale and no coherence has done a lot of maths and cannot use any of it; every question looks new. A student with coherence and no scale has beautiful principles and nothing to apply them to. The WHAT, HOW, WHY analysis from the Blueprint is precisely a coherence machine. It takes raw reps (scale) and files them under reaction circuits (coherence). That is why I attribute 90% of my 9.0 to it, and it is why the same machinery is what builds interview intuition.

3. Metacognition.

- The ability to think about your own thoughts.
- To observe your own thoughts from a third person perspective, as if reading out another individual's stream of consciousness.
- Being equipped with a multitude of mental models that are high yield for solving puzzles.

This is the skill the interview tests more than any exam ever could, because in the interview you will be wrong, in real time, in front of an expert. The question is never whether you will go down a dead end. You will. The question is how many seconds pass between you going down the dead end and you noticing.

A metacognitive student says, mid-sentence, "hold on, that cannot be right, because if n = 1 this gives a negative count". A non-metacognitive student keeps going until the interviewer stops them. The first student just demonstrated the single most valuable trait a supervisor wants in a student: they can be left alone with a problem and will not drift into nonsense for an hour.

The third bullet is the toolkit. Mental models are what your third-person observer reaches for when it spots a stall. Try a small case. Try the extreme case. Invert it. Is there symmetry? What is the converse? What breaks it? Draw it. They are enumerated in Section 3. They are not clever. They are reflexes, and reflexes are installed by repetition, not by reading about them once.

When these three skills are mixed together, you become capable of:

> **Articulating your non-verbal intuition in real time, while being able to spot that you are wrong.**

Read that sentence as three clauses and you will see the three skills in order. Articulating (vocalisation) your non-verbal intuition (intuition) while being able to spot that you are wrong (metacognition). That sentence is the North Star. Everything that follows is scaffolding to reach it.

Figure 1. The trinity.

Interview at the centre. Three branches, three sub-branches each. Three of them, one centre, which is why I call it the trinity. The pun on Trinity College is intentional; it is where I am applying, and this is the document I wish somebody had handed me. Whenever you train, point at the branch you are feeding.

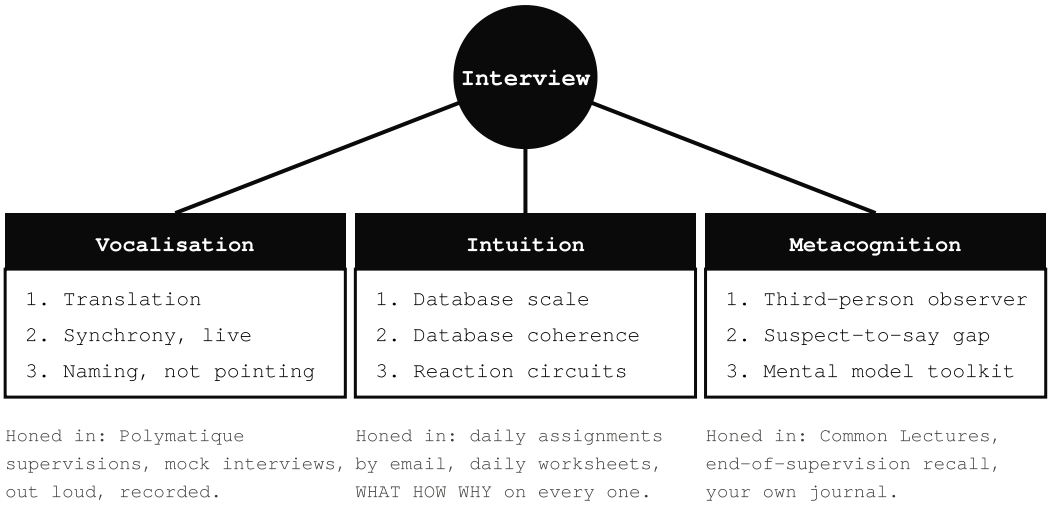


Figure 1. The trinity: three skills, one centre, and where each is honed in the TMUA9 and Polymatique stack.

The taxonomy. (Bloom's, drawn as a tree.)

Bloom's taxonomy is the standard pyramid of cognition: Knowledge at the base, then Comprehension, Application, Analysis, Evaluation, and at the very zenith, Creation. I draw it as a tree for a reason that is not decorative. A pyramid implies you stack blocks and stop. A tree implies roots that nobody sees, a trunk that everything grows through, layers that narrow as they rise, and a star at the top that only lights up when everything beneath it is alive. That is exactly how these skills behave.

The lower half of the tree (Knowledge, Comprehension, Application) is Intuition. It is the silent part. You either recognise the form of the question or you do not, and recognition is a database lookup.

The upper half (Analysis, Evaluation, Creation) is Vocalisation and Metacognition. This is the audible part. Analysis is where you break the problem apart and say what you are doing. Evaluation is where you judge your own path and spot that you are wrong. Creation is where you generalise: you take the lemma they gave you and extend it to a theorem they did not ask for. The star.

And underground, the roots. Intuition does not grow out of nothing. It grows out of two roots: raw intelligence and pattern recognition.

Raw intelligence stems from cognitive pain tolerance: the capacity to keep holding a problem in working memory while it hurts, while it refuses to resolve, while two ideas contradict, while you have been staring for four minutes and nothing has moved. Below that sits the ability to tolerate complexity of ideas: to let three partial structures coexist in your head without collapsing them prematurely into a wrong answer just to make the discomfort stop. This root is trained by sitting with hard problems for longer than feels reasonable. The Redprint's mistake number one (reading the model solution and thinking "ah, that makes sense") is an anaesthetic for cognitive pain. Stop taking it.

Pattern recognition stems from two things: database scaling and database coherence. Scaling is the raw count of problems, forms, tricks, counterexamples and proofs you have personally fought through. Coherence is the filing system: the degree to which those entries are linked by principle rather than sitting as isolated memories. Coherence is congruent to synthesis of ideas. When two entries that were filed apart suddenly become one entry because you saw the common mechanism, that is synthesis, and it is the only thing that makes a database usable at speed.

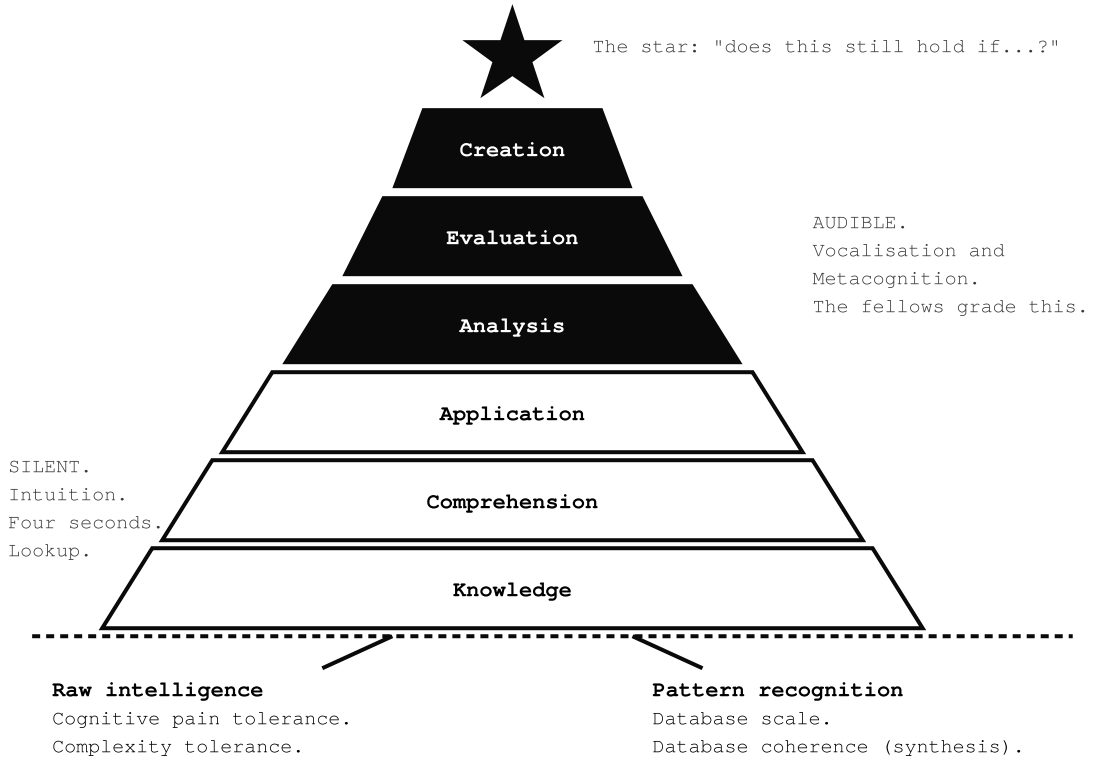


Figure 2. Bloom's taxonomy as a tree. Roots underground, intuition silent, the audible half above, the star on top.

Now read the tree bottom up, the way the interview actually runs. They hand you a question. Knowledge fires: you know what the objects are. Comprehension fires: you restate what is being asked. Application fires: a circuit suggests a tool. Up to here you have been mostly silent, and that is fine; this is the intuition half and it takes four seconds. Then Analysis: you start speaking,

breaking it into cases. Evaluation: you catch yourself, reroute, and say why. And if the room goes well, Creation: "actually, does this still hold if we replace 2 with any prime?" That last sentence is the one they remember when they write the report. It is also the only sentence in the room that was not a response to something they said. It is the first thing that was yours.

Where each pillar is honed.

Pillar	What trains it	Where
Intuition	Problem sheets, daily. Scale through reps, coherence through WHAT, HOW, WHY.	TMUA9 daily assignments by email, daily worksheets, the Discord.
Vocalisation	Mock interviews, out loud, mistakes left in. Then person to person: one supervisor, 5 to 8 students.	Mock interviews on the TMUA9 channel. Polymatique supervisions.
Metacognition	After every session: recall every statement made, by you and by the group. Truncate, leap, connect.	Common Lectures. The last 15 minutes of every supervision. Your own journal.

Leading a supervision out loud trains all three at once. That is the whole reason Polymatique exists.

Section 2: Attitude.

Last but never least is attitude. There is a common denominator of personality traits amongst those who excel in the room: curiosity, tenacity and passion. Concretely:

- Showing engagement and pure enthusiasm.
- Not being prideful about being able to solve problems.
- Curiosity: asking the interviewer questions about the working.
- Openness: being excited to completely destroy your own thoughts and come up with ideas that contradict your past arguments.
- Eradicating the urge to be arrogant when you get a question right, and diving into other methods to solve it.
- Having no guilt about being wrong in conjectures. You get to the answer with trials. In fact, you need to be wrong in the interview.

I want to expand on the last two, because they are where students bleed the most points without knowing it.

On arrogance. The moment you get a question right and visibly relax, you have told the interviewer that the answer was the point. For them the answer was never the point. A fellow who has set this question for eleven years does not care that you found 12. They care what you do in the second after you find 12. The student they want says "okay, that works, but it felt like a coincidence; is there a reason it had to be 12?" and goes looking for the second method. The second method is where mathematics lives. The first method is where the exam lives. You are not sitting an exam.

On guilt. Every conjecture you make out loud that turns out false is evidence that you make conjectures, and making conjectures is the job. A supervision at Trinity is one hour of somebody being wrong in front of a fellow and being corrected, every week, for three years. If you cannot be wrong comfortably in a 25 minute interview, they have reasonable doubt that you can do that. So I will say it the way I said the opposite thing in the Redprint: you need to be wrong in the interview. Not carelessly. Boldly. "I think this is always even. Let me check $n = 3$. No, it is odd. So it depends on the parity of n; let me see which way." That is a student who is going to be fine.

How to think of the interview.

You are neither in an inferior position nor a superior position to the interviewers. And once you get into Oxford or Cambridge, this is not how supervisions or tutorials are conducted at all. It is the sentiment Polymatique is built on: the supervisor is not at the centre teaching everything. The problem being solved is at the centre, and the supervisor is simply sitting in the crowd next to you, talking to you, and you are solving it together as a team.

Draw this geometry in your head before you walk in. Wrong picture: a desk, them on one side, you on the other, a question fired across it. Right picture: a whiteboard, the problem on it, three people standing in front of it, two of whom happen to have seen the problem before. When the interviewer gives you a hint, in the wrong picture it feels like you failed. In the right picture it is a teammate saying "try the small case". Take the hint with gratitude and run with it at full speed. Hints are not deductions. The students who got offers took hints; they just took them fast and did something with them.

One more thing about the picture. Because the problem is at the centre, you are allowed to talk to it. "So what is this thing actually asking me." "Why would anyone put a 6000 there." "That 2, 3, 5 looks deliberate." Talking to the problem is vocalisation that costs nothing and tells the interviewer that your engine is running.

Section 3: The Actual Blueprint to the Interview.

There are three things in this section. The mission statement, the generalisation machinery with worked examples, and the cadence and protocol.

[1] The mission statement when preparing.

ARTICULATE THE NON-VERBAL IN REAL TIME,
AND GENERALISE EVERYTHING I AM GIVEN.

[2] The generalisation machinery.

Great questions to ask back to the interviewer, starting with examples.

Example A. The Trinity specimen question.

In Trinity Maths Specimen Paper 1, they ask how many numbers in the closed interval [1, 6000] are multiples of none of 2, 3, 5. I came across Euler's totient function during my Putnam training, which turns this into a density argument using prime factorisation. This is a typical example of generalisation, so let me show the machinery firing on it, the way I showed a circuit firing in the Blueprint.

The exam answer: the numbers coprime to 30 recur with period 30, and there are $\varphi(30) = 8$ of them per period, so the count is $200 \times 8 = 1600$. Equivalently $6000\,(1 - 1/2)(1 - 1/3)(1 - 1/5) = 1600$. Fine.

The interview answer continues out loud: "That only worked because 30 divides 6000. What if the interval were [1, 5998]? Then the full periods still give exactly 8 per block, and all the error sits in the leftover block, so I need a floor correction: inclusion and exclusion with $\lfloor N/d \rfloor$ over the divisors d of 30. And what if instead of 2, 3, 5 it was any set of primes? Then the density of numbers coprime to their product P is $\varphi(P)/P$, and as I throw in more primes the product of $(1 - 1/p)$ goes to zero, which is the statement that almost no integer is coprime to everything." You just went from a counting question to the Euler product in ninety seconds, out loud, and every step was a question you asked yourself. That is the whole skill.

Example B. Polynomials and congruences.

If $a \equiv b \pmod m$, with m a natural number and a, b integers, then for any integer coefficient polynomial $p(x)$, $p(a) \equiv p(b) \pmod m$. After this you ask:

Is the converse true? If $p(a) \equiv p(b)$, are a and b always congruent? For true balance you need both balance and imbalance, so the opposite should also be validated. It is false in general: $p(x) = x^2$, $m = 5$, $a = 1$, $b = 4$ gives $p(a) \equiv p(b) \equiv 1$ but $1 \not\equiv 4$. Now the interesting question: for which p is the converse true? If p is injective modulo m, i.e. a permutation polynomial, it holds;

linear $p(x) = cx + d$ with $\gcd(c, m) = 1$ works. So the converse holds exactly when p permutes the residues. That is a boundary, and finding the boundary is the move.

Then iterate. Because a and b are just placeholders, $p(p(a)) \equiv p(p(b)) \pmod m$. One recursion means we can go to infinity: $p^{\circ k}(a) \equiv p^{\circ k}(b)$ for every k.

Then push it. What about exponential towers? Counterexample: $2 \equiv 7 \pmod 5$ but $2^2 = 4$ while $7^7 \equiv 3 \pmod 5$.

> **Even when you find counterexamples to a conjecture, there are specific conditions under which it is true. Find the boundary.**

Here is the boundary for the tower. $a^a \equiv b^b \pmod m$ fails because the exponent lives modulo $\varphi(m)$, not modulo m. So if you demand $a \equiv b \pmod m$ and also $a \equiv b \pmod{\varphi(m)}$, with $\gcd(a, m) = 1$, then $a^a \equiv b^b \pmod m$ by Euler's theorem. A tower of height n is controlled by the whole chain $m, \varphi(m), \varphi(\varphi(m)), \ldots, 1$. The conjecture was not wrong, it was under-conditioned. Say that sentence in the room.

Example C. The four statements.

If $p \Rightarrow q$ is proven, ask whether the converse $q \Rightarrow p$ is true. If it is hard to prove, try the contrapositive of the inverse, or the inverse of the contrapositive. These collapse into something neat. Write a statement as an ordered pair (p, q) meaning $p \Rightarrow q$. Then

- Converse: $C(p, q) = (q, p)$
- Inverse: $I(p, q) = (\neg p, \neg q)$
- Contrapositive: $CP(p, q) = (\neg q, \neg p)$

Compose them: $I(C(p, q)) = I(q, p) = (\neg q, \neg p) = CP(p, q)$, and $C(I(p, q)) = C(\neg p, \neg q) = (\neg q, \neg p) = CP(p, q)$. So $IC = CI = CP$. Each of C, I, CP is an involution, and together with the identity they form the Klein four-group $\{e, C, I, CP\}$. That is why there are exactly four statements and not eight, and why a statement and its contrapositive are logically equivalent while a statement and its converse are not: CP is the composite of the two operations, and equivalence is preserved by negating both sides, not by swapping them. This is a two minute thing to say out loud and it is exactly the kind of thing that makes an interviewer lean forward.

See how I was being metacognitive? I thought these thoughts, zoomed out, and then noted that I could express these thoughts. That is the loop. You can conjecture endlessly for generalisation. Usually there just needs to be one recurrence: when there is one connection between the n-th and the (n+1)-th, we can send it to infinity.

The one-step rule. If you can show that a property passes from stage n to stage n + 1, you have, for free, that it passes to every stage. Induction is the formal version. In the interview you do not need to write the induction; you need to say "this only used one step, so it goes all the way". Look for the one step in whatever they hand you.

The generalisation toolkit.

These are the reflexes the metacognitive observer reaches for. Install them so that they fire without being summoned. Every one has a vocal form: say the move name, then do the move. The name is for them; the move is for you.

Move	What you do	What you say
Small case	n = 0, 1, 2. Most structure is visible by n = 3.	"Let me try a small case."
Extreme case	$n \to \infty$, $x \to 0$, the degenerate triangle, the empty set. Boundaries expose the mechanism.	"What happens at the extreme."
Converse, inverse, contrapositive	Run the Klein four-group on every implication they give you. One of the four is usually the next question.	"Is the converse true?"
Relax a condition	They said integer; try rational. They said prime; try any m. Where it breaks is the theorem.	"Does this need n to be an integer?"
Tighten a condition	The conjecture failed; what extra hypothesis rescues it? (The tower above.)	"It was under-conditioned."
Change the number	Replace 2 with p, 6000 with N, the square with the k-th power. If the argument survives, you have generalised. If not, you found what the number was doing.	"Why 6000? Try N."
Go up a dimension	Points to lines, lines to planes, $\mathbb{R}$ to $\mathbb{R}^n$, numbers to polynomials.	"Does this work in 3D?"
Find the recurrence	Find the one step. Send it to infinity.	"This only used one step."
Invariant	What does the operation not change? Parity, sum, colouring, a modulus.	"What stays fixed here?"
Symmetry	Same under swapping a and b? Then the answer must be too. Halves the work and catches errors.	"This is symmetric in a and b."
Two methods	Solve it again differently. Ask why both arrive at the same place; that "why" is normally a deeper theorem.	"Is there a second way?"
Draw it	Stuck for twenty seconds with no picture on the board? Draw one.	"Let me draw this."

[3] The cadence and the protocol.

The cadence.

- **Vocalisation** is trained by solving out loud, every session, into a recording. Phone on the desk, voice memo running, one problem, speak every

thought as it occurs, including the stupid ones. Then listen back once. You will hate it. You will hear the 40 second silences and the "um, so, basically". That is the gap between Knowledge and Speaking, made audible. It closes in about two weeks of daily reps.

- **Synchrony** specifically: do not let yourself solve silently and then narrate. If you catch yourself going quiet, say "I have gone quiet, what I am thinking is..." and resume. That sentence is itself metacognition, and interviewers like hearing it.
- **Intuition** is trained by the Blueprint's WHAT, HOW, WHY on past interview questions, STEP questions and the Trinity specimen papers. Scale through reps, coherence through extraction. Every question goes into the journal under its principle, not under its topic. The daily TMUA9 assignments and worksheets are the rep stream.
- **Metacognition** is trained by the backtesting checklist from the Blueprint, run on interview-style problems, with one extra question: at which second did I first suspect I was wrong, and at which second did I say so? Close that gap. The Common Lectures teach the method; your own journal runs it.
- **Generalisation** is trained by never ending a problem at the answer. After every problem, one generalisation attempt, out loud, using one tool from the toolkit. Even if it fails. Especially if it fails, because then you look for the boundary.
- **Mock interviews**, with anyone who can hold a pen and stay quiet. They do not need to know maths. They need to stop you when you go silent and say "what are you thinking". One a week minimum. Record them.
- **Supervisions**. One Polymatique supervision a week: two hours, 5 to 8 applicants, one supervisor, problems solved out loud, recalled and analysed at the end. Leading one trains all three pillars at once.
- **Attitude** is trained by being wrong in front of people on purpose until it stops stinging. Post conjectures in the TMUA9 Discord. Let them get shot down. Say thank you.
- Never end a session on solving; end on one generalisation and one minute of listening to your own recording.

Interview day protocol.

1. On receiving the question: restate it in your own words, out loud, before touching the pen. (Comprehension, made audible.)
2. The first four seconds of silence are allowed; this is the intuition half firing. After that, speak. If nothing has fired, say "nothing is jumping out, so let me try a small case" and do it. The sentence buys you the time and shows the toolkit.
3. Narrate moves by name. "Small case." "Extreme case." "Let me check the converse." Then do them.
4. When you suspect you are wrong, say so in the same breath you suspect it. Do not finish the wrong line to see where it goes. "Hold on, that gives a

negative count, so I have made an error somewhere above."

5. When you receive a hint, say thank you with your work, not with words. Take it and go at full speed.
6. When you get the answer, do not stop. "That works. Is there a second way?" or "Does this still hold if..." One generalisation, every question, even if they move you on. They will move you on. Offer it anyway.
7. If you are completely stuck, say exactly what you are stuck on. "I can see I need to relate the exponent to m but I cannot see how." A precise statement of the stuck point is worth more than a vague attempt, because it tells them where your map ends.
8. Your emotions in the room are independent variables from your outcomes. Walking out feeling it went terribly means nothing. Execute in the thin slate of the present. Feel later.

What this is not.

- It is not a performance of confidence. Fellows have seen a thousand confident wrong students. They are looking for the correction reflex, not the posture.
- It is not a reading list. "Read around the subject" is the interview's version of "do more maths".
- It is not memorising past interview questions. You will never meet the same question. The aim, as always, is cognitive infrastructure: an algorithm you can run on whatever they put on the board, out loud.

Common pitfalls.

The Redprint lists all 31. The five that cost the most offers: going silent until you have the answer; saying "it is obvious"; treating a hint as a deduction; stopping at the answer; never listening back to your own recording. If you recognise yourself in one, that is the branch to feed this week.

Notes.

I started TMUA9 for the version of me that was sitting alone as a private candidate with no idea what any of this looked like from the inside. The TMUA part of that problem is solved by the Blueprint and the Redprint. The interview part is this document. The three skills are what I found when I asked why some students who could clearly do the maths got rejected, and some who could barely do it got offers. The ones who got offers could be heard thinking, could feel the right path before they could prove it, and could catch themselves. Vocalisation, intuition, metacognition. Everything above is scaffolding for those three words.

Everything referenced here is open-sourced at tmua9.com: the model answers for every TMUA paper in the Galleria, the Blueprint and the Redprint, free lectures on the TMUA9 YouTube channel, Polymatique supervisions at

polymatique.com, and the Discord where you can be wrong in public for practice. None of it is AI slop. Every PDF other than NOIR PREMIERE is free and will always stay free.

You do not need to be a genius to do any of this. You need a phone with a voice memo app, a problem, and the willingness to hear yourself be bad at it for two weeks. I am rooting for you from afar. May God be with you.

"But as for you, be strong and do not give up, for your work will be rewarded."
2 Chronicles 15:7 NIV

P.S. Print the trinity and the tree and pin them above the desk. Before each session, point at the branch you are about to feed. If you cannot point at one, you are about to do consumption, not preparation.

Best, Jamie. TMUA9.

9.