**How to Approach Solutions in Competitive Programming Key Techniques**

In my entire competitive programming (CP) journey, I’ve often encountered situations where a problem isn’t immediately clear, and finding a solution can feel like an uphill battle. We frequently hear about various algorithms and methods, but we rarely focus on the strategies needed to think clearly and apply them effectively. Here, I’ll share some techniques that have helped me approach and solve tough CP problems, and I hope they’ll be helpful to you as well.

So, you’ve read a problem and have no idea where to start. Here are some techniques that can often come in handy

**Technique 1 “Total Recall”**

Try to remember similar problems you’ve tackled before. Many CP problems don’t introduce completely new ideas; often, they are variations on familiar themes. You might be able to use your experience with a similar problem as a starting point.

**Technique 2 “From Specific to General”**

Sometimes it helps to solve specific cases before tackling the general problem. Simplifying a complex problem into smaller, manageable cases can reveal patterns or methods that you can scale up to the full solution.

Examples

* If the problem involves trees, try simplifying it to a straight path version.
* If the problem includes weights, consider cases where all weights are the same or where there are only a few distinct weights.

**Technique 3 “Bold Hypotheses”**

Don’t hesitate to make bold assumptions that feel intuitive. In a contest setting, trust your instincts and test out hypotheses quickly. Once you have one, try proving or disproving it with sample cases. Sometimes, bold ideas lead to breakthroughs.

Example Hypotheses

* Assume that a solution always exists within certain constraints.
* Hypothesize that there’s a limited number of states to handle.

**Technique 4 “Think Like the Problem”**

This might sound quirky, but it can be helpful to “step into” the problem’s scenario. Imagine that your role is to manage the input data, or simulate the actions needed. If the problem resembles a game, try visualizing or even playing it out. Visualizing solutions can often help with difficult problems.

**Technique 5 “Think Together”**

If you’re in a team contest, discussing simple observations with your teammates can spark new ideas. Statements like “If n is even, then the answer is 0” might lead to unexpected insights. Brainstorming with others can often help reveal the solution.

**Technique 6 “Pick a Method”**

Review popular algorithms and approaches and see if any might apply to the problem. Think about the problem’s constraints and limits, then try framing a solution using a particular approach, like divide and conquer, and build from there.

**Technique 7 “Print Out and Observe Patterns”**

Sometimes, writing a simple solution and printing its output across a range of inputs can reveal hidden patterns. Observing the output closely—sometimes even printed on paper—can help you spot useful patterns.

**Technique 8 “Research and Google”**

If allowed, using resources like the Online Encyclopedia of Integer Sequences (OEIS) or mathematical references online can help. Sometimes a quick search for a sequence or pattern can provide formal terms or insights to guide your approach.