

Understanding Problem-Solving Techniques

Understanding Problem-Solving Techniques

Study guide – concise, organized, and easy to read

1. What Is Problem Solving?

Aspect	Description
Definition	A systematic process for identifying a difficulty, generating possible solutions, evaluating alternatives, and implementing the best course of action.
Goal	Reach a satisfactory solution efficiently while learning from the process.
Scope	Applies to math, engineering, programming, business, daily life, and any situation that requires a decision or a fix.

2. General Stages of the Problem-Solving Process

1. Problem Identification

- Clarify **what** the problem is.
- Distinguish between symptoms and root causes.
- Write a clear problem statement.

2. Information Gathering

- Collect relevant data, constraints, and resources.
- Ask “who, what, where, when, why, how?”

3. Define Success Criteria

- List measurable objectives (e.g., speed, cost, accuracy).
- Prioritize criteria (must-have vs. nice-to-have).

4. Generate Options (Ideation)

- Use brainstorming, mind maps, or analogical thinking.
- Aim for quantity first; defer judgment.

Understanding Problem-Solving Techniques

5. Analyze & Select

- Apply decision tools (pros/cons, cost?benefit, weighted scoring).
- Check feasibility against constraints and success criteria.

6. Plan Implementation

- Break the chosen solution into tasks.
- Assign responsibilities, set timelines, and anticipate risks.

7. Execute & Monitor

- Follow the plan, collect real?time feedback.
- Adjust if deviations appear.

8. Review & Reflect

- Evaluate outcomes vs. criteria.
- Document lessons learned for future problems.

3. Core Problem?Solving Techniques

Technique	When to Use	Key Steps	Example
Trial?and?Error	Simple, low?cost problems; limited options	1. Test a solution 2. Observe result 3. Keep or discard	Guessing a combination lock
Divide?and?Conquer (Decomposition)	Complex problems with many parts	1. Break problem into sub?problems 2. Solve each independently 3. Reassemble	Sorting large data set via merge sort
Algorithmic / Procedural	Repetitive tasks, well ?defined inputs/outputs	1. Define step?by?step procedure 2. Follow strictly 3. Validate each step	Euclidean algorithm for GCD
Heuristics	Large search spaces where optimal solution is costly	1. Apply rule?of?thumb (e.g., "nearest neighbor") 2. Iterate quickly 3. Refine if needed	Approximate traveling?salesman route

Understanding Problem-Solving Techniques

Technique	When to Use	Key Steps	Example
Analogical Reasoning	When a new problem resembles a known one	1. Identify a similar past problem 2. Map its solution onto current context 3. Adapt	Using electrical circuit analysis to model fluid flow
Root Cause Analysis (RCA)	Persistent or recurring issues	1. Ask "Why?" repeatedly (5 Whys) 2. Build cause-effect diagram (Ishikawa) 3. Target fundamental cause	Fixing recurring software crashes
Brainstorming & SCAMPER	Creative or open-ended challenges	1. Generate ideas without judgment 2. Use SCAMPER prompts (Substitute, Combine, ...)	