🧠 What To Do When You See a Truly New Pattern

Think of this as **Plan B** when recall fails.

🔑 The Golden Truth (Read this carefully)

> **All "new" patterns are just combinations or re-framings of known ideas.**
> You are not discovering a new law of physics — you are discovering a new *view*.

🪜 The 4-Step Pattern Discovery Mantra

Use this when:

- You don't recognize the pattern
- You feel blank
- You've never seen this problem type before

1️⃣ Shrink the Problem (Make It Small)

Ask:

> "What happens for n = 1, 2, 3?"

Example:

- 1 element
- 2 elements

- 3 elements

New patterns emerge only with **tiny inputs**.

2️⃣ Write the Brute Force Explicitly

Say out loud:

> "If I had infinite time, I would…"

Examples:

- Check all pairs
- Check all subarrays
- Try all paths
- Try all combinations

Now ask:

> "What part is repeating?"

The repeated work is the key.

3️⃣ Name the Repeated Work

Repeated work almost always falls into one of these buckets:

Repeated thing	**Known technique**
Rechecking sums	Prefix sum

Rechecking ranges	Sliding window
Rechecking nearest greater/smaller	Monotonic stack
Rechecking ordering	Sorting + greedy
Rechecking connectivity	DFS / BFS
Rechecking pair conditions	Hashing
Rechecking subarray boundaries	Two pointers
Rechecking combinations	DP / bitmask

This is where **new patterns collapse into known ones**.

4️⃣ Define an Invariant (This Creates the Pattern)

Ask:

> “What must always be true while I scan?”

Examples:

- “Window sum ≤ K”
- “Stack is monotonic.”
- “Prefix remainder seen before.”

- “Left half always sorted.”
- “Distance from origin fixed.”

If you can state an invariant, **you have a pattern**.

🧠 How Patterns Are Actually Born (Important Insight)

Many famous patterns were once “new”:

- Sliding Window was “prefix sum optimization.”
- Monotonic Stack was “nearest element scanning.”
- Two pointers were used in a “bidirectional scan.”
- Binary search on the answer was a “decision problem.”

They were discovered by:

> Shrinking → brute force → observing repetition → enforcing invariant

You can do the same.

🚨 What NOT to Do When You See a New Pattern

❌ Panic
❌ Jump to YouTube / LeetCode solution immediately
❌ Try to memorize the solution
❌ Assume you are “weak.”

Even top engineers pause when they see a new structure.

🎯 Interview-Safe Answer If You're Truly Stuck

If this happens in an interview, say:

> "Let me start with the brute force approach to understand the structure, then optimize."

This is a **correct and respected response**.

🧘‍♂️ Confidence Reset (Important)

You already:
✔ Understand core patterns
✔ Know invariants
✔ Know how to reason

So even if the pattern is unfamiliar:

> **Your process still works.**

Patterns are tools — reasoning is the skill.

🔥 Final One-Line Mantra for New Patterns

> **Shrink → brute → find repetition → enforce invariant**

Write this somewhere visible.