

## Day 24 Backtracking

### ITSRUNTYM

### What is Backtracking?

**Backtracking** is a **recursive technique** to build a solution incrementally, **backtrack** as soon as you determine that the current path won't lead to a valid solution.

#### Key Idea

- Try a possibility.
- If it doesn't work, backtrack and try the next one.
- Repeat until a solution is found or all possibilities are exhausted.

#### Steps to Solve Using Backtracking

1. **Choose**: Choose an option from a set of valid options.
2. **Explore**: Recurse to explore further using that choice.
3. **Unchoose**: Undo the choice (backtrack), and try another option.

#### When to Use Backtracking?

Use backtracking when:

- Problem requires **all combinations/permutations/subsets**.
- Problem involves **decision making at each step**, e.g., yes/no, pick/skip.
- You see **constraints** like "must not repeat", "must be unique", or "follow specific rules".
- You're asked to **generate all solutions** or **find a valid one** among them.

#### Common Examples

Problem	Type
N-Queens	Constraint satisfaction
Sudoku Solver	Constraint satisfaction
Subsets / Combinations	Combinatorics
Word Search in Grid	Path finding
Permutations	Combinatorics
Rat in a Maze	Path finding
Knight's Tour	Chess-based puzzle





### How Backtracking Works:










Let's understand with a **Subset Generation** problem:






Given: nums = [1, 2]

Each level: Consider including or excluding a number

Backtracking explores both choices.

#	Problem	Asked In	Description	LeetCode Link
1	<b>N-Queens</b>	Amazon, Microsoft	Place N queens on an NxN board so that no two queens threaten each other.	 <a href="#">N-Queens</a>
2	<b>Sudoku Solver</b>	Google, Apple	Solve a 9x9 Sudoku board using backtracking and validation.	 <a href="#">Sudoku Solver</a>
3	<b>Word Search</b>	Facebook, Amazon	Search if a word exists in a 2D board by exploring adjacent cells.	 <a href="#">Word Search</a>
4	<b>Letter Combinations of a Phone Number</b>	Google, Meta	Generate all letter combinations using T9 keypad rules.	 <a href="#">Letter Combinations</a>

5	<b>Subsets (Power Set)</b>	Amazon, Microsoft	Return all possible subsets of a set of distinct numbers.	 <a href="#">Subsets</a>
6	<b>Permutations / Permutations II</b>	Apple, Amazon	Generate all possible permutations (with/without duplicates).	 <a href="#">Permutations</a> /  <a href="#">Permutations II</a>
7	<b>Combinations / Combination Sum</b>	Facebook, Google	Return combinations that sum to target (with or without reuse).	 <a href="#">Combinations</a> /  <a href="#">Combination Sum</a> /  <a href="#">Combination Sum II</a>
8	<b>Palindrome Partitioning</b>	Microsoft, Google	Partition a string into all possible palindrome substring groups.	 <a href="#">Palindrome Partitioning</a>
9	<b>Rat in a Maze</b>	Flipkart, Amazon	Find all possible paths in a grid from top-left to bottom-right.	 <a href="#">GFG Rat in a Maze</a> (on GFG)
10	<b>Knight's Tour</b>	Amazon	Move a knight over a chessboard visiting all cells exactly once.	 <a href="#">GFG Knight's Tour</a> (variation on GFG)

11	<b>Restore IP Addresses</b>	Meta, Google	Restore all valid IP addresses from a digit-only string.	 <a href="#">Restore IP Addresses</a>
12	<b>Generate Parentheses</b>	Facebook, Amazon	Generate all valid combinations of n pairs of parentheses.	 <a href="#">Generate Parentheses</a>
13	<b>Binary Watch</b>	Google	Return all possible times a binary watch could represent with n LEDs.	 <a href="#">Binary Watch</a>
14	<b>Expression Add Operators</b>	Google	Add operators + - * between digits to reach a target value.	 <a href="#">Expression Add Operators</a>
15	<b>Matchsticks to Square</b>	Amazon	Determine if matchsticks can form a square (NP-complete type).	 <a href="#">Matchsticks to Square</a>