

- ChatGPT
- Explore GPTs
- Today
- Recursion Problem List
- HashMap Interview Questions
- Top Graph Problems
- Math Interview Problems
- String Problems Difficulty Levels
- Array Problems Summary
- Merge Intervals Example
- Fast and Slow Pointers
- Matrix Pattern Problems
- Two Pointers Problems

backtracking based top 15 problem commonly asked in interview questions.
give input and output of the problem.
list 5 easy, 5 medium, 5 hard difficulty problems.



Memory updated

Here's a list of commonly asked backtracking problems categorized by difficulty:

Easy

1. Subset Sum

- **Problem:** Given a set of integers, find all possible subsets (the power set) of the set.
- **Input:** `[1, 2]`
- **Output:** `[], [1], [2], [1, 2]`



Get citation

2. Permutations of a String

- **Problem:** Generate all possible permutations of a string.
- **Input:** `"abc"`
- **Output:** `["abc", "acb", "bac", "bca", "cab", "cba"]`

3. Combination Sum (Unique Combinations)

- **Problem:** Find all unique combinations of numbers that sum up to a target value.
- **Input:** `candidates = [2, 3, 6, 7], target = 7`
- **Output:** `[[2, 2, 3], [7]]`

4. Letter Combinations of a Phone Number

- **Problem:** Given a digit string, return all possible letter combinations that the number could represent.
- **Input:** `"23"`
- **Output:** `["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"]`



5. Generate Parentheses

- **Problem:** Generate all combinations of well-formed parentheses.
- **Input:** `n = 3`
- **Output:** `["((()))", "(()())", "(())()", "()(())", "()()()"]`

Medium

1. N-Queens

- **Problem:** Place N queens on an N x N chessboard so that no two queens attack each other.
- **Input:** `4`
- **Output:** `[[".Q...", "...Q", "Q...", "...Q."], [".Q.", "Q...", "...Q", ".Q..."]]`

2. Combination Sum



- **Problem:** Find all possible combinations of candidates that add up to a target sum. Candidates can be used multiple times.
- **Input:** `candidates = [2, 3, 5], target = 8`
- **Output:** `[[2, 2, 2, 2], [2, 3, 3], [3, 5]]`

3. Subsets

- **Problem:** Given a set of distinct integers, return all possible subsets.

- **Input:** `[1, 2, 3]`
- **Output:** `[[], [1], [2], [1, 2], [3], [1, 3], [2, 3], [1, 2, 3]]`

4. Word Search II

- **Problem:** Given a board and a list of words, find all words in the board.
- **Input:** `board = [{"o","a","a","n"}, {"e","t","a","e"}, {"i","h","k","r"}, {"i","f","l","v"}], words = ["oath","pea","eat","rain"]`
- **Output:** `["eat", "oath"]`



5. Restore IP Addresses

- **Problem:** Given a string containing only digits, restore it by returning all possible valid IP address combinations.
- **Input:** `"25525511135"`
- **Output:** `["255.255.11.135", "255.255.111.35"]`

Hard

1. Word Break II

- **Problem:** Given a string and a dictionary of words, add spaces in the string to construct a sentence where each word is a valid dictionary word.
- **Input:** `s = "catsanddog", wordDict = ["cat", "cats", "and", "sand", "dog"]`
- **Output:** `["cat sand dog", "cats and dog"]`

2. Sudoku Solver

- **Problem:** Solve a Sudoku puzzle by filling the empty cells. A valid solution must satisfy the Sudoku constraints.
- **Input:** `board = [{"5","3",".", ".", ".", "7", ".", ".", ".", "."}, {"6",".", ".", ".", "1","9","5",".", ".", ".", "."}, {".", ".", "9","8",".", ".", ".", ".", "6","."}, {".", ".", ".", ".", "6",".", ".", ".", "3"}, {".", ".", ".", ".", "4",".", ".", ".", ".", "."}, {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}, {".", ".", ".", ".", ".", ".", ".", ".", ".", "."}]`
- **Output:** Solved board.

3. All Valid Palindromic Partitions

- **Problem:** Given a string, partition it such that every substring is a palindrome. Return all possible palindromic partitions.
- **Input:** `"aab"`
- **Output:** `[["a", "a", "b"], ["aa", "b"]]`

4. Combination Sum III


- **Problem:** Find all possible combinations of k numbers that add up to a number n, where the combinations are unique and the numbers used in the combination are from 1 to 9.
- **Input:** `k = 3, n = 7`
- **Output:** `[[1, 2, 4]]`

5. N-Queens II

- **Problem:** Solve the N-Queens problem and return the number of distinct solutions.
- **Input:** `4`
- **Output:** `2`

These problems span a range of complexities and are often used to test understanding of backtracking and combinatorial algorithms.

Two different models

 **Upgrade plan**
Get GPT-4, DALL-E, and more

 Message ChatGPT



ChatGPT can make mistakes. Check important info.

