

- 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

dynamic programming 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.

Sure! Here's a list of common dynamic programming problems categorized by difficulty, along with their typical inputs and outputs:

### Easy

#### 1. Climbing Stairs

- Problem:** You are climbing a staircase. It takes `n` steps to reach the top. Each time you can either climb 1 or 2 steps. How many distinct ways can you climb to the top?
- Input:** `n = 4`
- Output:** `5` (Ways: 1+1+1+1, 1+1+2, 1+2+1, 2+1+1, 2+2)

Get citation

#### 2. House Robber

- Problem:** You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, but adjacent houses have security systems connected. If two adjacent houses are robbed, the alarm will go off. Find the maximum amount of money you can rob tonight without alerting the police.
- Input:** `nums = [2, 3, 2]`
- Output:** `3` (Robbing only the second house)

#### 3. Maximum Subarray

- Problem:** Find the contiguous subarray within a one-dimensional array of numbers that has the largest sum.
- Input:** `nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]`
- Output:** `6` (Subarray `[4, -1, 2, 1]`)

#### 4. Fibonacci Number

- Problem:** Given `n`, calculate the `n`th Fibonacci number.
- Input:** `n = 5`
- Output:** `5` (Fibonacci sequence: 0, 1, 1, 2, 3, 5)

#### 5. Min Cost Climbing Stairs

- Problem:** You are given an array `cost` where `cost[i]` is the cost of the `i`th step on a staircase. Once you pay the cost, you can either climb one or two steps. Find the minimum cost to reach the top of the staircase.
- Input:** `cost = [10, 15, 20]`
- Output:** `15` (Minimum cost: 15 from steps `[10, 5]`)

### Medium

#### 1. Longest Increasing Subsequence

- Problem:** Given an integer array `nums`, return the length of the longest strictly increasing subsequence.
- Input:** `nums = [10, 9, 2, 5, 3, 7, 101, 18]`
- Output:** `4` (Longest increasing subsequence: `[2, 3, 7, 101]`)

#### 2. Edit Distance

- Problem:** Given two strings `word1` and `word2`, return the minimum number of operations required to convert `word1` to `word2`. Operations include insertions, deletions, and substitutions.

- **Input:** `word1 = "horse", word2 = "ros"`
- **Output:** `3` (Operations: `horse`  $\rightarrow$  `ros` by delete 'h', 'r', and substitute 'e' for 's')

### 3. Partition Equal Subset Sum

- **Problem:** Given a non-empty array `nums`, determine if it can be partitioned into two subsets such that the sum of elements in both subsets is the same.
- **Input:** `nums = [1, 5, 11, 5]`
- **Output:** `true` (Can be partitioned into `[1, 5, 5]` and `[11]`)



### 4. Unique Paths II

- **Problem:** A robot is located at the top-left corner of a `m x n` grid. The robot can only move right or down, but some cells are blocked. Find the number of unique paths to the bottom-right corner.
- **Input:** `obstacleGrid = [[0,0,0],[0,1,0],[0,0,0]]`
- **Output:** `2` (Two unique paths)

### 5. Decode Ways

- **Problem:** A message containing letters from `A-Z` can be encoded into numbers using the mapping `A = 1`, `B = 2`, ..., `Z = 26`. Given a string `s` consisting of digits, return the number of ways to decode it.
- **Input:** `s = "12"`
- **Output:** `2` (Decodings: "AB", "L")

## Hard

### 1. Longest Palindromic Substring



- **Problem:** Given a string `s`, return the longest palindromic substring in `s`.
- **Input:** `s = "babad"`
- **Output:** `"bab"` (or `"aba"`)

### 2. Word Break II

- **Problem:** Given a string `s` and a dictionary of strings `wordDict`, return all possible sentences that can be formed by segmenting `s` into a space-separated sequence of one or more dictionary words.
- **Input:** `s = "catsanddog"`, `wordDict = ["cat", "cats", "and", "sand", "dog"]`
- **Output:** `["cats and dog", "cat sand dog"]`

### 3. Longest Valid Parentheses

- **Problem:** Given a string containing just the characters `'('` and `')'`, find the length of the longest valid (well-formed) parentheses substring.
- **Input:** `s = "(()())"`
- **Output:** `6`



### 4. Minimum Window Substring

- **Problem:** Given two strings `s` and `t`, return the minimum window in `s` which will contain all the characters in `t` in any order.
- **Input:** `s = "ADOBECODEBANC"`, `t = "ABC"`
- **Output:** `"BANC"`

### 5. Interleaving String

- **Problem:** Given three strings `s1`, `s2`, and `s3`, return true if `s3` is formed by the interleaving of `s1` and `s2`.
- **Input:** `s1 = "aabcc"`, `s2 = "dbbca"`, `s3 = "aadbcbccac"`
- **Output:** `true`

Is this conversation helpful so far?



Two premium features



Upgrade plan

Get GPT-4, DALL-E, and more



Message ChatGPT



ChatGPT can make mistakes. Check important info.

