

Beginner-level Dynamic Programming problems:

1. Nth Fibonacci Number

Problem Description:

Find the Nth Fibonacci number using dynamic programming.

Input Format:

- Integer n

Output Format:

- Integer representing the Nth Fibonacci number

Constraints:

- $0 \leq n \leq 30$

Example:

Input: $n = 5$

Output: 5

Explanation: Fibonacci sequence: 0, 1, 1, 2, 3, 5...

Practice Link: [GeeksforGeeks - Nth Fibonacci Number](#)

2. Climbing Stairs

Problem Description:

Count the number of distinct ways to climb to the top of a staircase with n steps. You can climb 1 or 2 steps at a time.

Input Format:

- Integer n

Output Format:

- Integer representing number of ways

Constraints:

- $1 \leq n \leq 44$

Example:

Input: $n = 4$

Output: 5

Explanation: Ways: [1,1,1,1], [1,1,2], [2,1,1], [1,2,1], [2,2]

Practice Link: [GeeksforGeeks - Count Ways to Reach Nth Stair](#)

3. Min Cost Climbing Stairs

Problem Description:

Given an array `cost[]`, where `cost[i]` is the cost to step on the i th stair, find the minimum cost to reach the top.

Input Format:

- Array `cost[]`

Output Format:

- Integer representing minimum cost

Constraints:

- $2 \leq \text{cost.length} \leq 10^5$
- $0 \leq \text{cost}[i] \leq 999$

Example:

Input: `cost = [10, 15, 20]`

Output: 15

Explanation: Start at index 1, pay 15, reach top.

Practice Link: [GeeksforGeeks - Min Cost Climbing Stairs](#)

4. House Robber

Problem Description:

Given an array `nums` representing money in each house, find the maximum amount you can rob without robbing adjacent houses.

Input Format:

- Array `nums[]`

Output Format:

- Integer representing maximum money

Constraints:

- $(1 \leq \text{nums.length} \leq 100)$
- $(0 \leq \text{nums}[i] \leq 400)$

Example:

Input: `nums = [2,7,9,3,1]`

Output: 12

Explanation: Rob houses 1, 3, and 5 $\rightarrow 2 + 9 + 1 = 12$

Practice Link: [LeetCode - House Robber](#)

5. Count Ways to Reach Nth Stair (1, 2, or 3 steps)

Problem Description:

Count the number of ways to reach the `n`th stair using steps of size 1, 2, or 3.

Input Format:

- Integer `n`

Output Format:

- Integer representing number of ways

Constraints:

- $(1 \leq n \leq 44)$

Example:

Input: `n = 4`

Output: 7

Explanation: All combinations using 1, 2, or 3 steps

Practice Link: [GeeksforGeeks - Count Ways Using 1, 2, 3 Steps](#)

6. Coin Change (Number of Ways)

Problem Description:

Given coins of different denominations and a total amount, find the number of combinations to make up that amount.

Input Format:

- Array `coins[]`, Integer `sum`

Output Format:

- Integer representing number of combinations

Constraints:

- $(1 \leq \text{sum} \leq 10^3)$
- $(1 \leq \text{coins}[i] \leq 10^4)$

Example:

Input: `coins = [1, 2, 3]`, `sum = 4`

Output: 4

Explanation: [1,1,1,1], [1,1,2], [2,2], [1,3]

Practice Link: [GeeksforGeeks - Coin Change](#)

7. Subset Sum Problem

Problem Description:

Given an array and a target sum, determine if there is a subset whose sum equals the target.

Input Format:

- Array `arr[]`, Integer `sum`

Output Format:

- Boolean (`true` or `false`)

Constraints:

- Array of positive integers

Example:

Input: `arr = [3, 34, 4, 12, 5, 2]`, `sum = 9`

Output: `true`

Explanation: Subset [4, 5] sums to 9

Practice Link: [GeeksforGeeks - Subset Sum Problem](#)

8. Longest Common Subsequence (LCS)

Problem Description:

Given two strings, find the length of their longest common subsequence.

Input Format:

- Strings s_1, s_2

Output Format:

- Integer representing length of LCS

Constraints:

- $(1 \leq \text{length} \leq 10^3)$

Example:

Input: $s_1 = \text{"ABCDGH"}, s_2 = \text{"AEDFHR"}$

Output: 3

Explanation: LCS is "ADH"

Practice Link: [GeeksforGeeks - LCS](#)

9. Longest Palindromic Subsequence

Problem Description:

Given a string, find the length of the longest subsequence that is a palindrome.

Input Format:

- String s

Output Format:

- Integer representing length of longest palindromic subsequence

Constraints:

- $1 \leq \text{length} \leq 1000$

Example:

Input: s = "bbabcbcab"

Output: 7

Explanation: "babcbab" is the longest palindromic subsequence

Practice Link: [GeeksforGeeks - Longest Palindromic Subsequence](#)