

Homework 5 : Dynamic Programming(DP)

1. Climbing Stairs

Problem:

You are climbing a staircase. It takes n steps to reach the top. You can climb either 1 step or 2 steps at a time.

Count how many distinct ways you can climb to the top.

Input:

$n = 5$

Output:

8

2. Min Cost Climbing Stairs

Problem:

Given an array $cost$ where $cost[i]$ is the cost of step i , return the minimum cost to reach the top. You can climb either 1 or 2 steps.

Input:

$cost = [10, 15, 20]$

Output:

15

3. House Robber

Problem:

You are a robber. Each house has some money. You can't rob two adjacent houses. Return the maximum amount you can rob.

Input:

$nums = [2, 7, 9, 3, 1]$

Output:

12

4. Fibonacci Number

Problem:

Return the `n`-th Fibonacci number.

Input:

`n = 7`

Output:

13

5. Max Sum of Non-Adjacent Elements

Problem:

Given an array, return the maximum sum of non-adjacent elements.

Input:

`arr = [3, 2, 5, 10, 7]`

Output:

15

6. Jump Game I

Problem:

Given an array `nums`, where each element represents your max jump from that position, determine if you can reach the end.

Input:

`nums = [2, 3, 1, 1, 4]`

Output:

True

7. Jump Game II

Problem:

Find the minimum number of jumps to reach the end of the array.

Input:

```
nums = [2, 3, 1, 1, 4]
```

Output:

```
2
```

8. Coin Change

Problem:

You are given coins of different denominations and a total amount. Find the minimum number of coins to make that amount. Return -1 if not possible.

Input:

```
coins = [1, 2, 5], amount = 11
```

Output:

```
3
```

9. Decode Ways

Problem:

A message containing letters from A-Z is encoded with numbers '1' to '26'. Count the total number of ways to decode it.

Input:

```
s = "226"
```

Output:

```
3
```

10. Longest Increasing Subsequence

Problem:

Find the length of the longest strictly increasing subsequence in an array.

Input:

```
nums = [10, 9, 2, 5, 3, 7, 101, 18]
```

Output:

```
4
```