

Advanced Programming

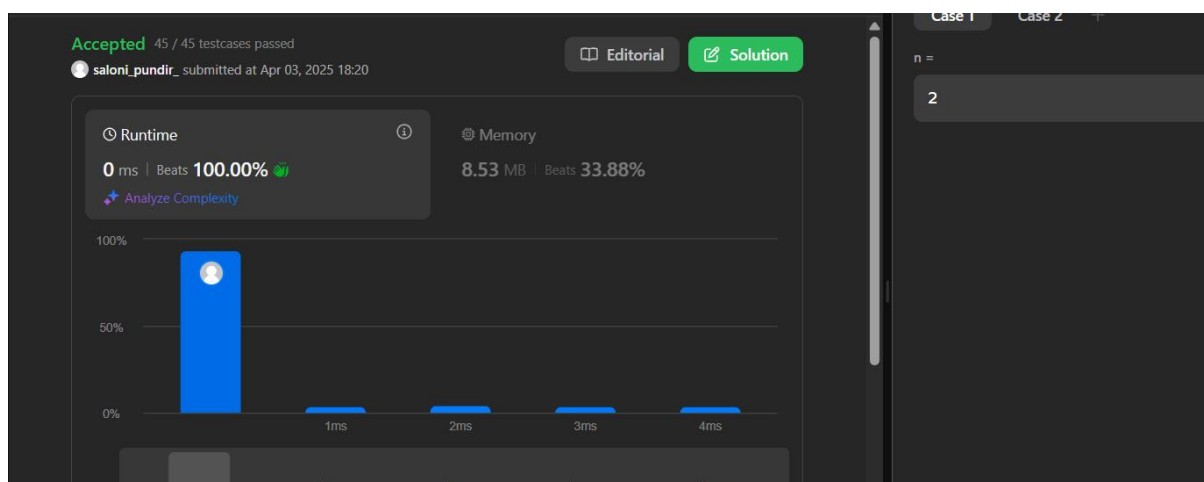
ASSIGNMENT 07

Q1. Climbing Stairs

Code:

```
C++ v Auto
1  class Solution {
2  public:
3      int solve(int n,vector<int> &dp){
4          //base case
5          if(n<=2)
6              return n;
7
8          if(dp[n]!=-1)
9              return dp[n];
10
11         dp[n]=solve(n-1,dp)+solve(n-2,dp);
12         return dp[n];
13     }
14     int climbStairs(int n) {
15         if(n<=2)
16             return n;
17         vector<int> dp(n+1,-1);
18         return solve(n,dp);
19     }
20 };
```

Output:



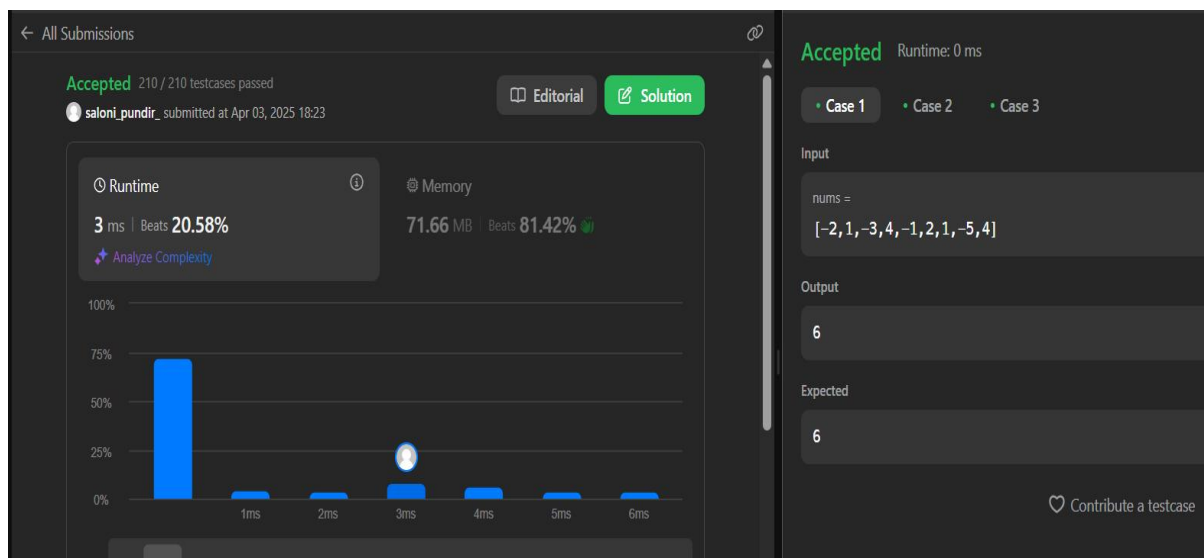
Q2. Maximum Subarray

Code:

```
C++ v Auto

1  class Solution {
2  public:
3      int maxSubArray(vector<int>& nums) {
4          int res = nums[0];
5          int total = 0;
6
7          for (int n : nums) {
8              if (total < 0) {
9                  total = 0;
10             }
11
12             total += n;
13             res = max(res, total);
14         }
15
16         return res;
17     }
18 };
```

Output:



Q3. House Robber.

Code:

```
C++  Auto

1  class Solution {
2  public:
3      int rob(vector<int>& nums) {
4          int n = nums.size();
5
6          if (n == 1) {
7              return nums[0];
8          }
9
10         vector<int> dp(n, 0);
11
12         dp[0] = nums[0];
13         dp[1] = max(nums[0], nums[1]);
14
15         for (int i = 2; i < n; i++) {
16             dp[i] = max(dp[i - 1], nums[i] + dp[i - 2]);
17         }
18
19         return dp[n - 1];
20     }
21 };
```

Output:

← All Submissions

Accepted 70 / 70 testcases passed

saloni_pundir, submitted at Apr 03, 2025 18:26

Editorial Solution

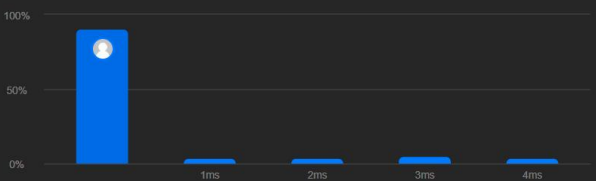
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

10.60 MB | Beats 61.79%



Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =

[1,2,3,1]

Output

4

Expected

4

Contribute a testcase

Q4. Jump Game.

Code:

```
C++  Auto

1  class Solution {
2  public:
3      bool canJump(vector<int>& nums) {
4          int goal = nums.size() - 1;
5
6          for (int i = nums.size() - 2; i >= 0; i--) {
7              if (i + nums[i] >= goal) {
8                  goal = i;
9              }
10         }
11
12         return goal == 0;
13     }
14 };|
```

Output:

