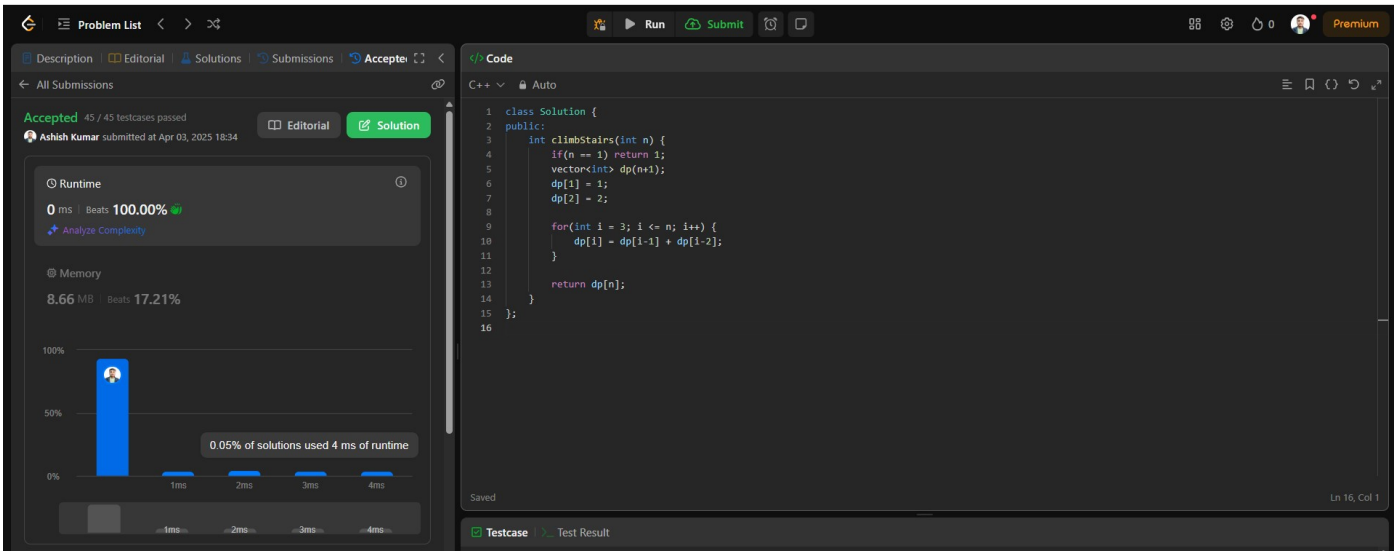
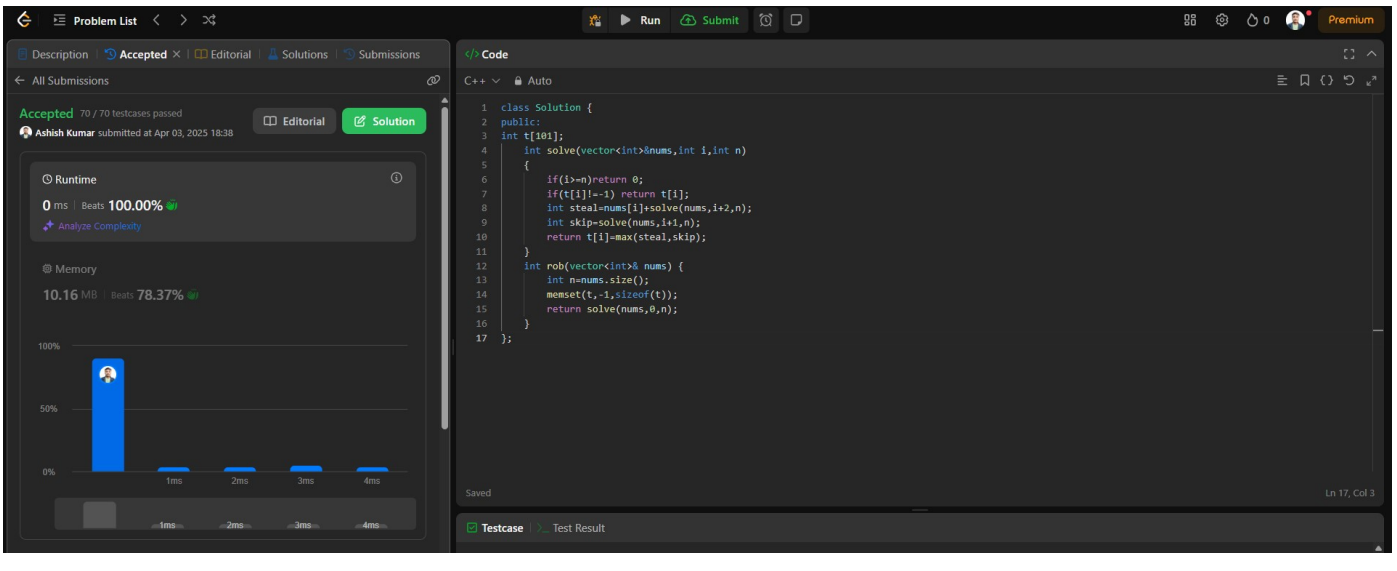
Experiment 7

Student Name: Ashish Kumar **UID:22bcs11958**
Branch: CSE **Section/Group:614(B)**
Semester: 6 **Date of Performance:03/04/25**
Subject Name: AP Lab **Subject Code: 22CSP-351**

```cpp
class Solution {
public:
    int climbStairs(int n) {
        if(n == 1) return 1;
        vector<int> dp(n+1);
        dp[1] = 1;
        dp[2] = 2;

        for(int i = 3; i <= n; i++) {
            dp[i] = dp[i-1] + dp[i-2];
        }

        return dp[n];
    }
};
```

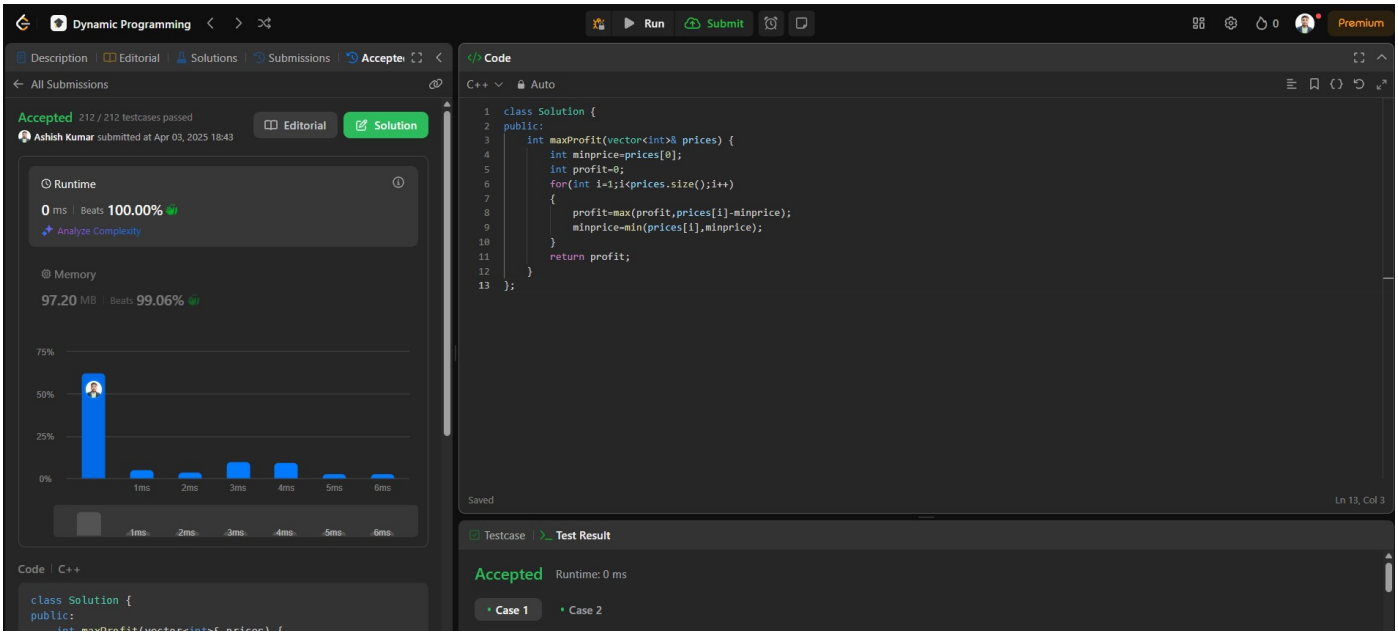


```cpp
class Solution {
public:
int t[101];
    int solve(vector<int>&nums,int i,int n)
    {


        if(i>=n)return 0;
        if(t[i]!=-1) return t[i];
        int steal=nums[i]+solve(nums,i+2,n);
        int skip=solve(nums,i+1,n);
        return t[i]=max(steal,skip);
    }
    int rob(vector<int>& nums) {
        int n=nums.size();
        memset(t,-1,sizeof(t));
        return solve(nums,0,n);
    }
};
```

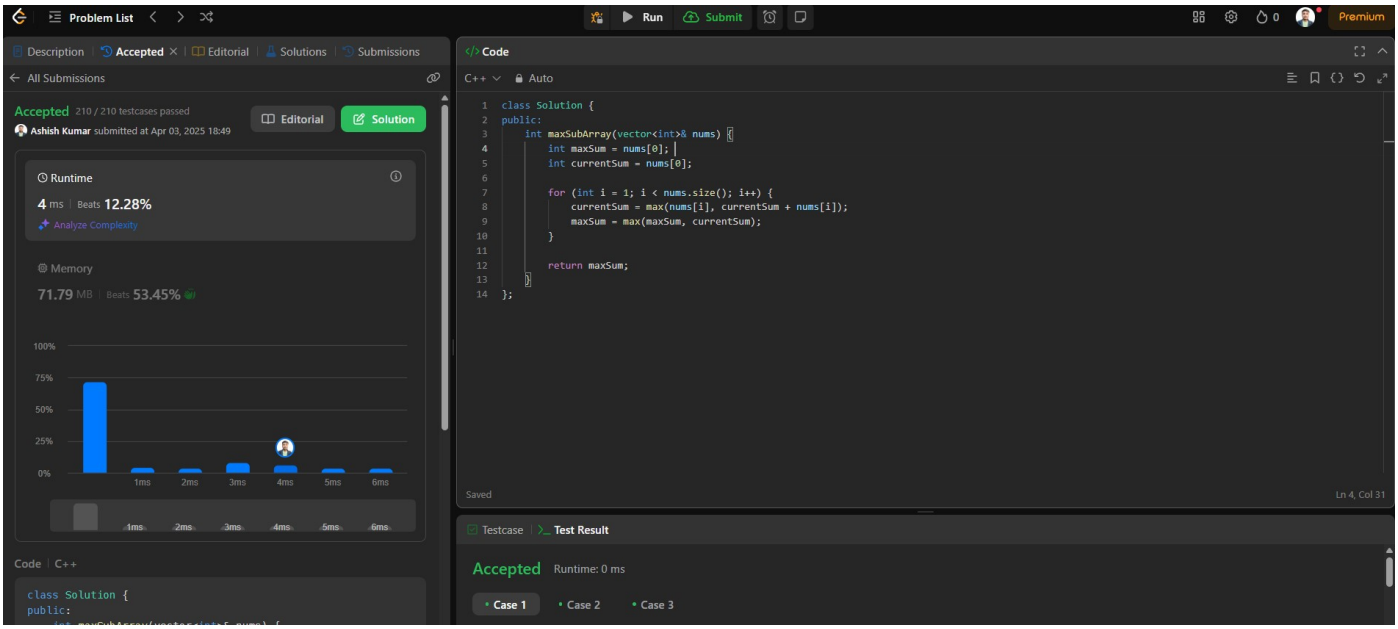



```cpp
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        int minprice=prices[0];
        int profit=0;
        for(int i=1;i<prices.size();i++)
        {
            profit=max(profit,prices[i]-minprice);
            minprice=min(prices[i],minprice);
        }
```

```
        return profit;
    }
};
```




```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0];
        int currentSum = nums[0];

        for (int i = 1; i < nums.size(); i++) {
            currentSum = max(nums[i], currentSum + nums[i]);
            maxSum = max(maxSum, currentSum);
        }

        return maxSum;
    }
};
```


Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 210 / 210 testcases passed
Ashish Kumar submitted at Apr 03, 2025 18:49
Editorial
Solution
Runtime
4 ms | Beats 12.28%
Analyze Complexity
Memory
71.79 MB | Beats 53.45%
100%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
5ms
6ms
Code | C++
class Solution {
public:
Code
C++
Auto
class Solution {
public:
int maxSubArray(vector<int>& nums) {
int maxSum = nums[0];
int currentSum = nums[0];
for (int i = 1; i < nums.size(); i++) {
currentSum = max(nums[i], currentSum + nums[i]);
maxSum = max(maxSum, currentSum);
}
return maxSum;
}
};
Saved
Ln 4, Col 31
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3