

Pratistha Tewari

22BCS13955

614-B

1. Climbing Stairs:

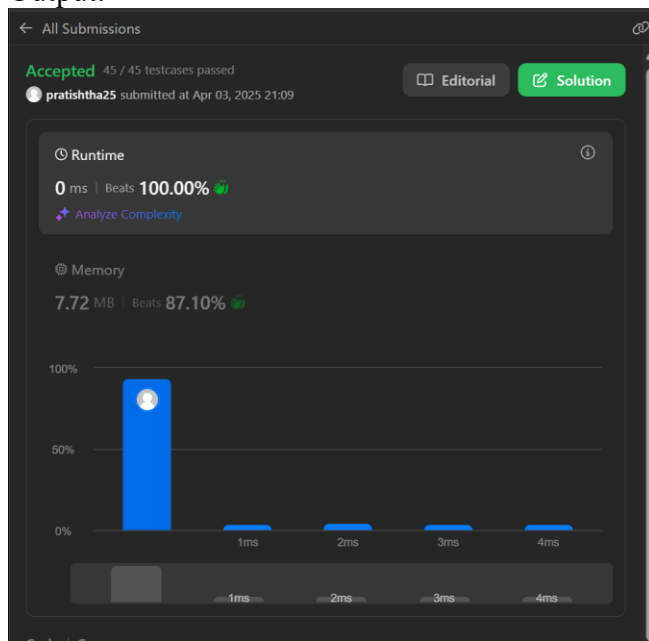
Code:

```
class Solution {
public:
    int climbStairs(int n) {
        if (n == 1) return 1;
        if (n == 2) return 2;

        int prev1 = 1, prev2 = 2, curr;
        for (int i = 3; i <= n; i++) {
            curr = prev1 + prev2;
            prev1 = prev2;
            prev2 = curr;
        }

        return prev2;
    }
};
```

Output:



2. Maximum Subarray

Code:

```
#include <vector>
#include <algorithm>
```

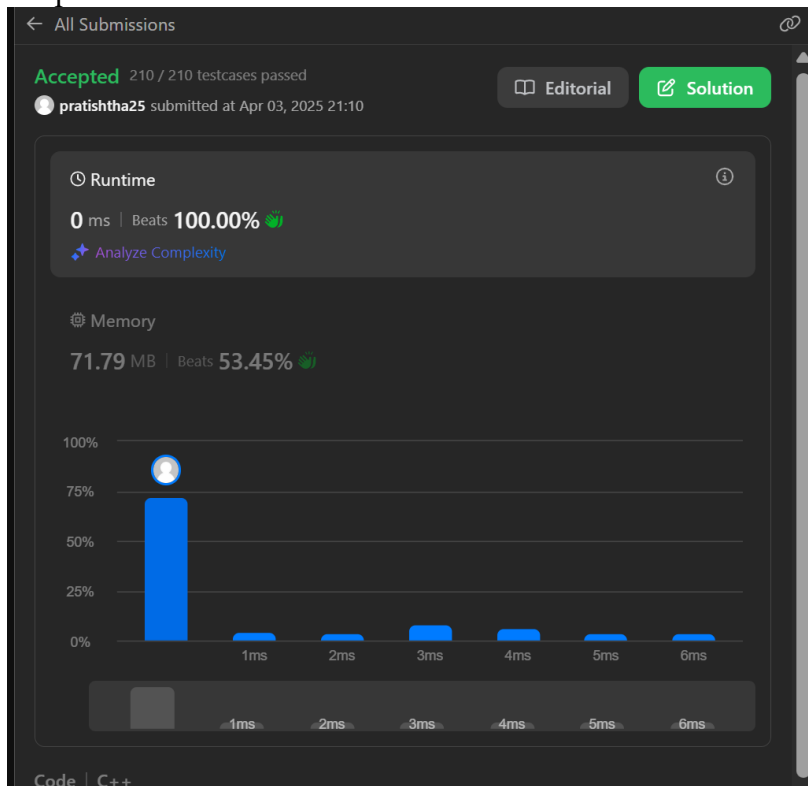
```
using namespace std;
```

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

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

        return maxSum;
    }
};
```

Output:



3. House Robber

Code:

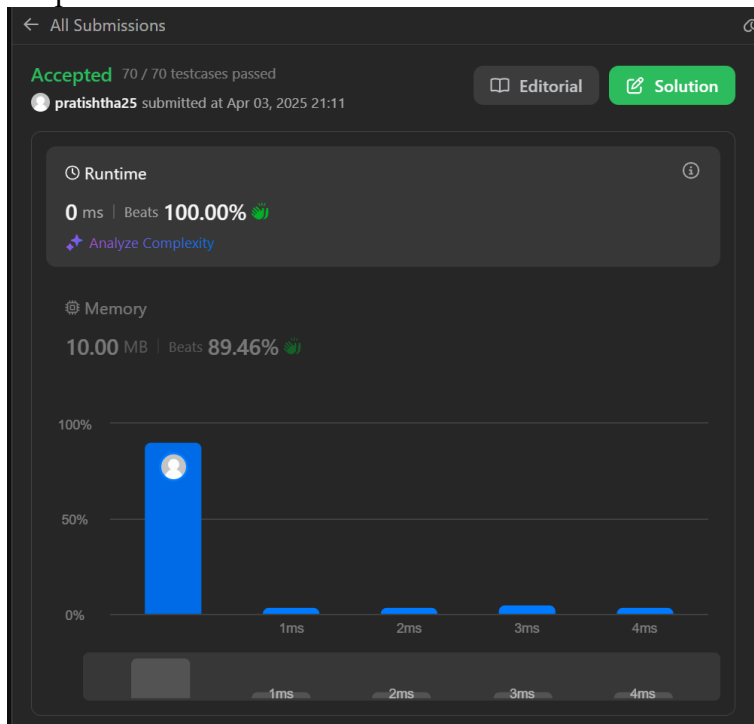
```
class Solution {
public:
    int rob(vector<int>& nums) {
        int n = nums.size();
        if (n == 0) return 0;
        if (n == 1) return nums[0];

        int prev2 = 0, prev1 = nums[0];

        for (int i = 1; i < n; i++) {
            int take = nums[i] + prev2;
            int skip = prev1;
            int curr = max(take, skip);
            prev2 = prev1;
            prev1 = curr;
        }

        return prev1;
    }
};
```

Output:



4. Jump Game

Code:

```
#include <vector>
```

```
using namespace std;
```

```
class Solution {
```

```
public:
```

```
    bool canJump(vector<int>& nums) {  
        int maxReach = 0; // Farthest index we can reach
```

```
        for (int i = 0; i < nums.size(); i++) {  
            if (i > maxReach) return false; // If we can't reach index i, return false  
            maxReach = max(maxReach, i + nums[i]); // Update max reach  
        }
```

```
        return true;
```

```
    }
```

```
};
```

Output:

