

NAME : Yashita

UID : 22BCS15024

SECTION : FI\_Iot 601 'A'

Ap experiment 7

### 1. Climbing Stairs

```
class Solution {
```

```
public:
```

```
    int climbStairs(int n) {
```

```
        vector<int> dp(n + 1);
```

```
        dp[0] = 1;
```

```
        dp[1] = 1;
```

```
        for (int i = 2; i <= n; ++i) {
```

```
            dp[i] = dp[i - 1] + dp[i - 2];
```

```
        }
```

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

```
    }
```

```
};
```

← All Submissions

Accepted 45 / 45 testcases passed

Yashitaa submitted at Apr 06, 2025 23:42

Editorial

Solution

Runtime

0 ms | Beats 100.00% 🏆

Analyze Complexity

Memory

8.65 MB | Beats 17.38%

100%



C++ v Auto

```
1 class Solution {
2 public:
3     int climbStairs(int n) {
4         vector<int> dp(n + 1);
5         dp[0] = 1;
6         dp[1] = 1;
7
8         for (int i = 2; i <= n; ++i) {
9             dp[i] = dp[i - 1] + dp[i - 2];
10        }
11    }
```

Saved

Testcase | Test Result

Accepted

Runtime: 0 ms

Case 1

Case 2

## 2. [Jump Game](#)

```
class Solution {  
  
public:  
  
    bool canJump(vector<int>& nums) {  
  
        int i = 0;  
  
        for (int reach = 0; i < nums.size() && i <= reach; ++i)  
            reach = max(reach, i + nums[i]);  
  
        return i == nums.size();  
    }  
};
```

The screenshot shows a code editor interface for a C++ solution. On the left, there's a sidebar with 'All Submissions' and a status bar indicating 'Accepted' with 174/174 testcases passed. Below this, a 'Runtime' section shows '3 ms' and 'Beats 28.48%', and a 'Memory' section shows '52.04 MB' and 'Beats 99.17%'. A progress bar is also visible. The main editor area shows the C++ code for the 'canJump' function. On the right, there's a 'Testcase' section showing 'Accepted' with a runtime of 0 ms, and two cases are listed: 'Case 1' and 'Case 2'.

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

## 3. [Maximum Product Subarray](#)

```
class Solution {  
  
public:  
  
    int maxProduct(vector<int>& nums) {  
  
        int ans = nums[0];  
  
        int dpMin = nums[0];  
  
        int dpMax = nums[0];  
  
        for (int i = 1; i < nums.size(); ++i) {
```

```

const int num = nums[i];

const int prevMin = dpMin;

const int prevMax = dpMax;

if (num < 0) {

    dpMin = min(prevMax * num, num);

    dpMax = max(prevMin * num, num);

} else {

    dpMin = min(prevMin * num, num);

    dpMax = max(prevMax * num, num);

}

ans = max(ans, dpMax);

}

return ans;

}

};

```

← All Submissions



C++ v Auto

Accepted 190 / 190 testcases passed

Yashitaa submitted at Apr 06, 2025 23:45

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

17.75 MB | Beats 57.88%



```

1 class Solution {
2 public:
3     int maxProduct(vector<int>& nums) {
4         int ans = nums[0];
5         int dpMin = nums[0];
6         int dpMax = nums[0];
7
8         for (int i = 1; i < nums.size(); ++i) {
9             const int num = nums[i];
10            const int prevMin = dpMin;
11            const int prevMax = dpMax;
12            if (num < 0) {
13                dpMin = min(prevMax * num, num);

```

Saved

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =

#### 4. Perfect Squares

```
class Solution {  
  
public:  
  
    int numSquares(int n) {  
  
        vector<int> dp(n + 1, n);  
  
        dp[0] = 0;  
  
        dp[1] = 1;  
  
  
        for (int i = 2; i <= n; ++i)  
  
            for (int j = 1; j * j <= i; ++j)  
  
                dp[i] = min(dp[i], dp[i - j * j] + 1);  
  
  
        return dp[n];  
  
    }  
};
```

← All Submissions

Accepted 589 / 589 testcases passed  
Yashitaa submitted at Apr 06, 2025 23:45

Editorial Solution

Runtime  
35 ms | Beats 89.52%  
Analyze Complexity

Memory  
13.20 MB | Beats 66.59%

30%  
20%  
10%

C++ Auto

```
4 vector<int> dp(n + 1, n);  
5 dp[0] = 0;  
6 dp[1] = 1;  
7  
8 for (int i = 2; i <= n; ++i)  
9     for (int j = 1; j * j <= i; ++j)  
10         dp[i] = min(dp[i], dp[i - j * j] + 1);  
11  
12 return dp[n];  
13 }  
14 }  
15
```

Saved

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input