

NAME : Anish Patial

UID : 22BCS15029

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

Anish Patial submitted at Apr 06, 2025 22:22

Editorial

Solution

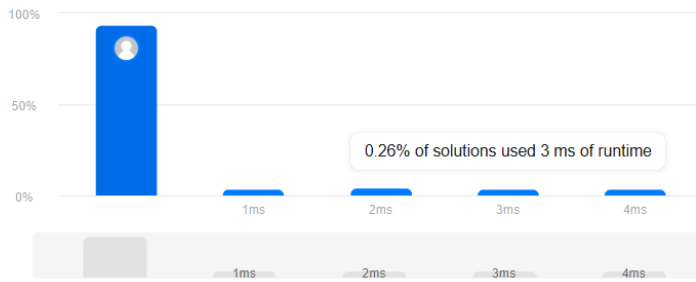
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

8.62 MB | Beats 17.38%



Code | C++

C++ 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
12        return dp[n];
13    }
14 }
15 };
```

Saved

Testcase Test Result

Accepted Runtime: 0 ms

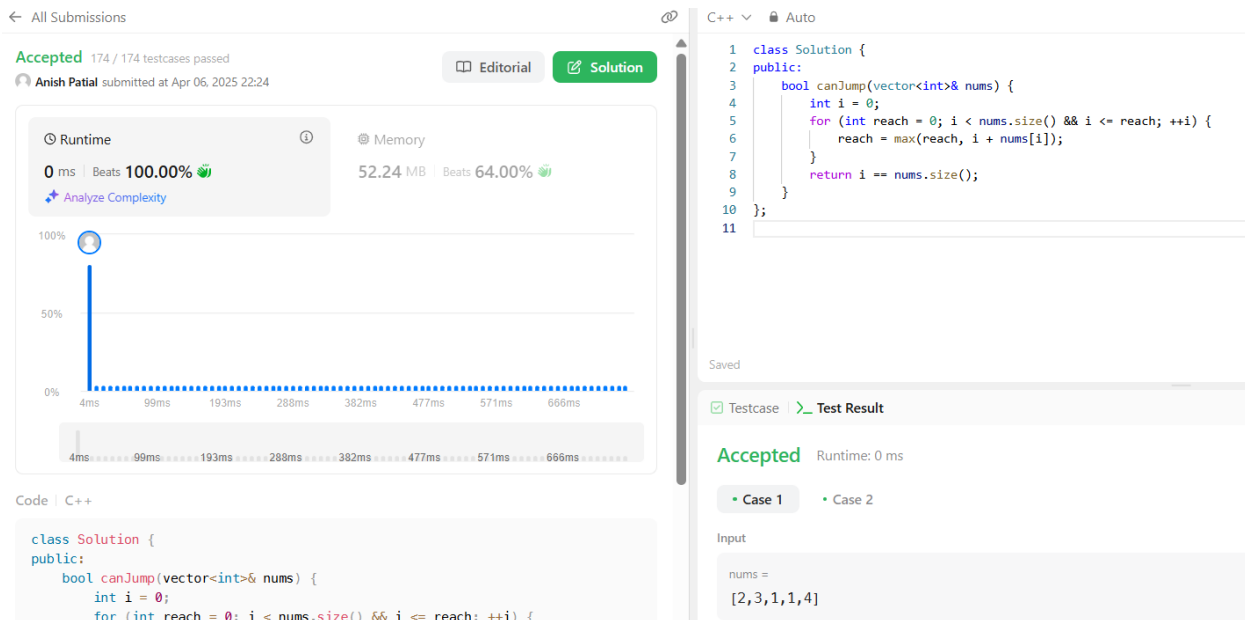
Case 1 Case 2

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();
    }
};
```



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);
            }
        }
        return ans;
    }
};
```

```

    }

    ans = max(ans, dpMax);

}

return ans;

}

};

```

Accepted 190 / 190 testcases passed

Anish Patial submitted at Apr 06, 2025 22:25

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

17.59 MB | Beats 98.74%



Code | C++

```
class Solution {
```

```

10     const int prevMin = dpMin;
11     const int prevMax = dpMax;
12     if (num < 0) {
13         dpMin = min(prevMax * num, num);
14         dpMax = max(prevMin * num, num);
15     } else {
16         dpMin = min(prevMin * num, num);
17         dpMax = max(prevMax * num, num);
18     }
19     ans = max(ans, dpMax);
20 }
21
22 return ans;
23 }
24 };
25

```

Saved

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1

Case 2

Input

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];

```

```

}

```

```

};

```

Accepted 589 / 589 testcases passed

Anish Patil submitted at Apr 06, 2025 22:26

Editorial

Solution

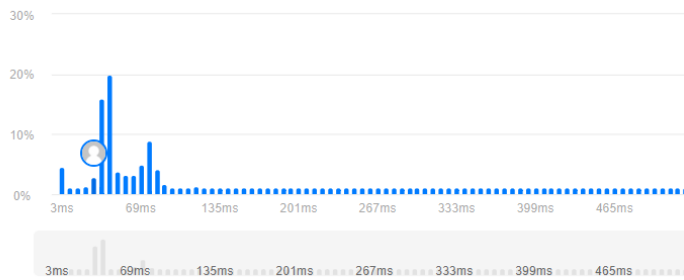
Runtime

32 ms | Beats 91.19%

Analyze Complexity

Memory

13.03 MB | Beats 82.81%



Code | C++

```

1 class Solution {
2 public:
3     int numSquares(int n) {
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