



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 6

Student Name: Samarth Yadav

Branch: CSE

Semester: 6th

Subject Name: Advanced Programming - 2

UID: 22BCS13134

Section/Group: 637-B

Date of Performance: 27/2/25

Subject Code: 22CSH-351

Ques 1:

Aim: Climbing Stairs

Code:

class Solution(object):

def climbStairs(self, n):

if n <= 2:

return n

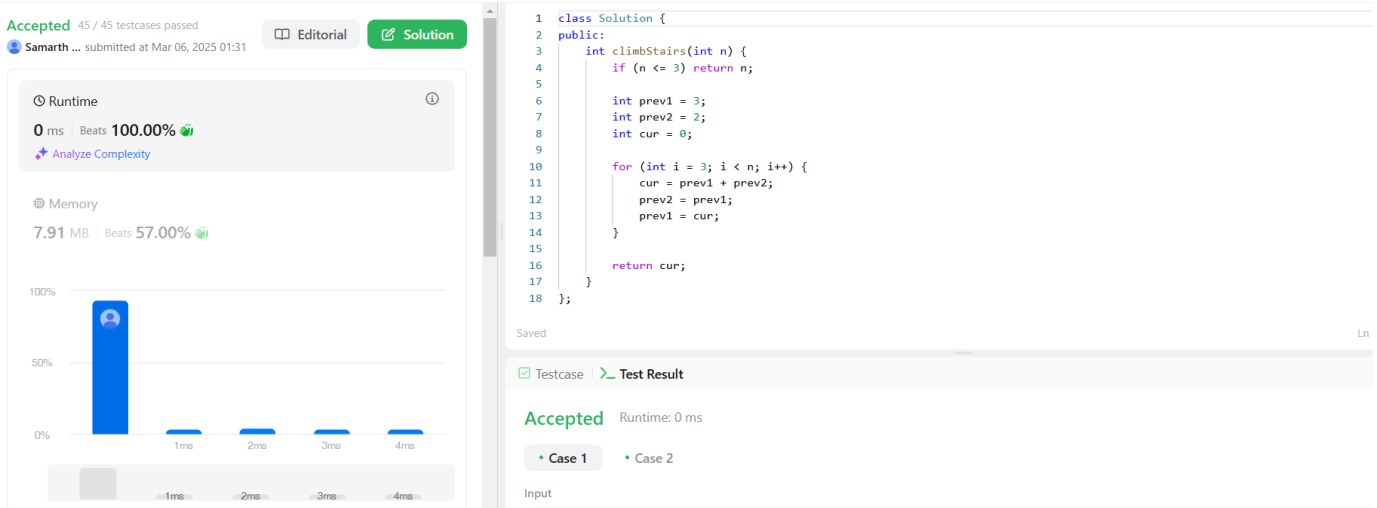
a, b = 1, 2

for _ in range(3, n + 1):

a, b = b, a + b

return b

Submission Screenshot:



Submission Link:

<https://leetcode.com/problems/climbing-stairs/submissions/1564163110/>



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Ques 2:

Aim: Maximum Subarray

Code:

```
class Solution(object):  
    def maxSubArray(self, nums):  
        max_sum = cur_sum = nums[0]  
        for num in nums[1:]:  
            cur_sum = max(num, cur_sum + num)  
            max_sum = max(max_sum, cur_sum)  
        return max_sum
```

Submission Screenshot:

Accepted 210 / 210 testcases passed

Samarth Yadav submitted at Aug 15, 2024 19:38

Editorial

Solution

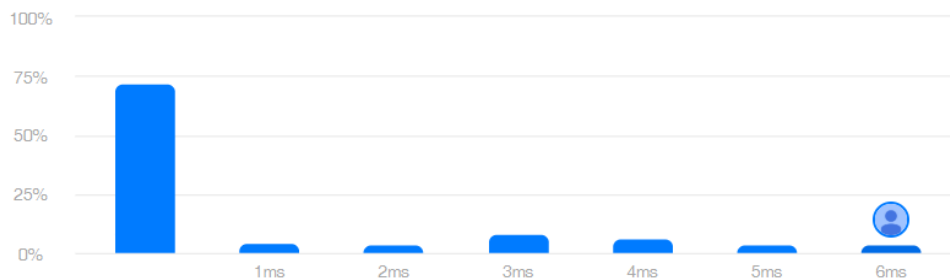
Runtime

71 ms | Beats 5.09%

Analyze Complexity

Memory

70.54 MB | Beats 99.97%



Submission Link:

<https://leetcode.com/problems/maximum-subarray/submissions/1356647094/>



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

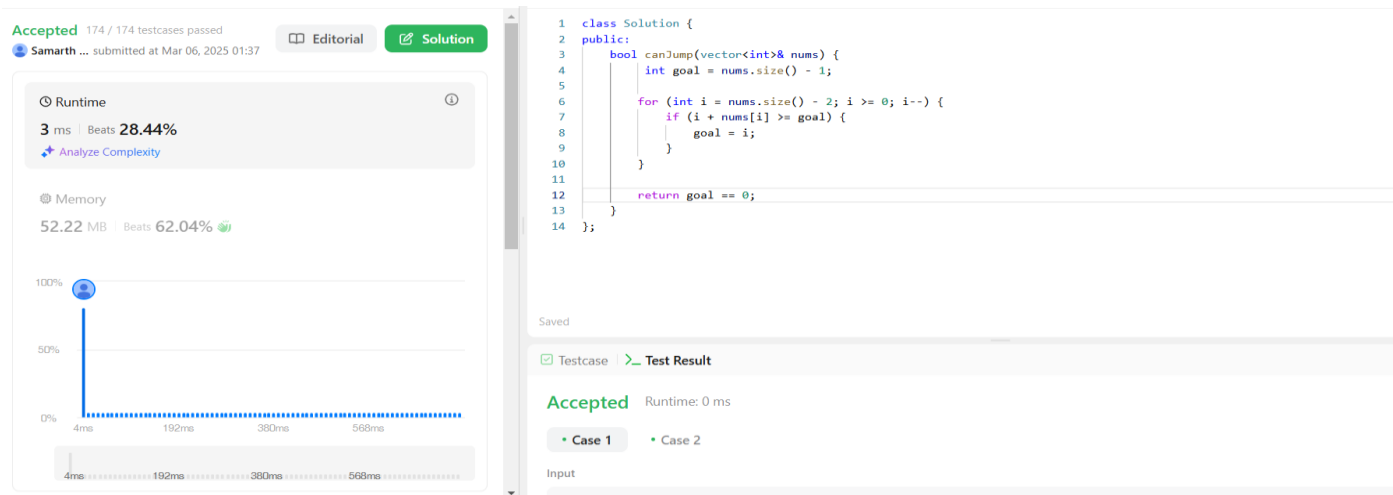
Ques 3:

Aim: Jump Game

Code:

```
class Solution(object):  
    def canJump(self, nums):  
        max_reach = 0  
        for i, num in enumerate(nums):  
            if i > max_reach:  
                return False # If we reach an index we cannot jump to  
            max_reach = max(max_reach, i + num)  
        if max_reach >= len(nums) - 1:  
            return True # If we can reach the last index  
        return False
```

Submission Screenshot:



Submission Link:

<https://leetcode.com/problems/jump-game/submissions/1564167231/>