



WORKSHEET 6

Student Name: Parth Arora

UID: 22BCS16661

Branch: BE-CSE

Section/Group: 22BCS_NTPP-602-A

Semester: 6th

Date of Performance: 20/02/2025

Subject Name: AP LAB - II

Subject Code: 22CSP-351

1. **Aim:** You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

2. Source Code:

```
class Solution:
    def climbStairs(self, n: int) -> int:
        if n == 0 or n == 1:
            return 1
        prev, curr = 1, 1
        for i in range(2, n+1):
            temp = curr
            curr = prev + curr
            prev = temp
        return curr
```

3. Screenshots of outputs:

The screenshot displays a code editor interface for a submission. On the left, the submission status is 'Accepted' with 45/45 testcases passed. The runtime is 0 ms (Beats 100.00%) and memory usage is 17.94 MB (Beats 8.26%). The right pane shows the Python code for the Solution class, which implements the climbStairs method. Below the code, the 'Testcase' tab is selected, showing 'Case 1' with input 'n = 2' and output '2'.

2.

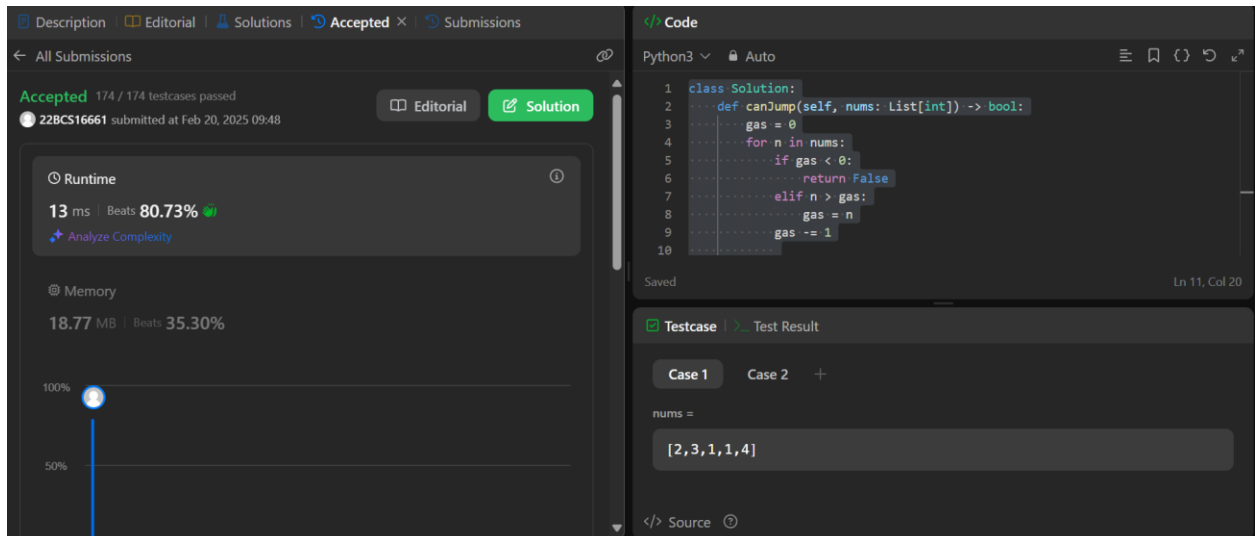
Aim: You are given an integer array `nums`. You are initially positioned at the array's **first index**, and each element in the array represents your maximum jump length at that position. Return true *if you can reach the last index*, or false *otherwise*.

Source Code:

```
class Solution:
    def canJump(self, nums: List[int]) -> bool:
        gas = 0
        for n in nums:
            if gas < 0:
                return False
            elif n > gas:
                gas = n
            gas -= 1

        return True
```

Screenshots of outputs:



The screenshot displays a code editor interface for a problem solution. The top navigation bar includes tabs for Description, Editorial, Solutions, Accepted (selected), and Submissions. Below this, the submission status is 'Accepted' with 174/174 testcases passed, submitted by user 228CS16661 on Feb 20, 2025 at 09:48. The code is written in Python3 and is saved. The runtime is 13 ms, beating 80.73% of solutions, and the memory usage is 18.77 MB, beating 35.30% of solutions. A test case is shown with `nums = [2, 3, 1, 1, 4]`.

```
1 class Solution:
2     def canJump(self, nums: List[int]) -> bool:
3         gas = 0
4         for n in nums:
5             if gas < 0:
6                 return False
7             elif n > gas:
8                 gas = n
9             gas -= 1
10
```

3.

Aim: Given an integer array `nums`, find a subarray that has the largest product, and return *the product*. The test cases are generated so that the answer will fit in a **32-bit** integer.

Source Code:

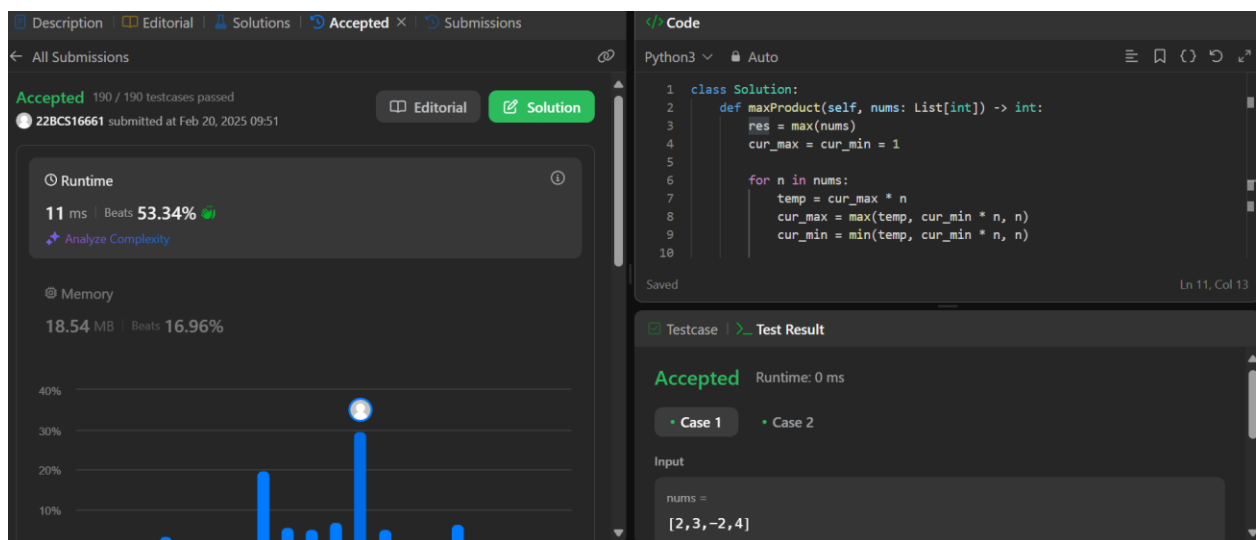
```
class Solution:
    def maxProduct(self, nums: List[int]) -> int:
        res = max(nums)
        cur_max = cur_min = 1

        for n in nums:
            temp = cur_max * n
            cur_max = max(temp, cur_min * n, n)
            cur_min = min(temp, cur_min * n, n)

        res = max(res, cur_max)

        return res
```

4. Screenshots of outputs:





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.