**Experiment 6**

**Student Name : Payal UID: 22BCS15824**

**Branch: CSE Section/Group: 637-B**

**Semester: 6th Date of Performance:27/2/25**

**Subject Name: Advanced Programming - 2 Subject Code: 22CSH-351**

**Ques 1:**

**Aim:** Climbing Stairs

**Code:**

class Solution(object):

def climbStairs(self, n):

if n <= 2:

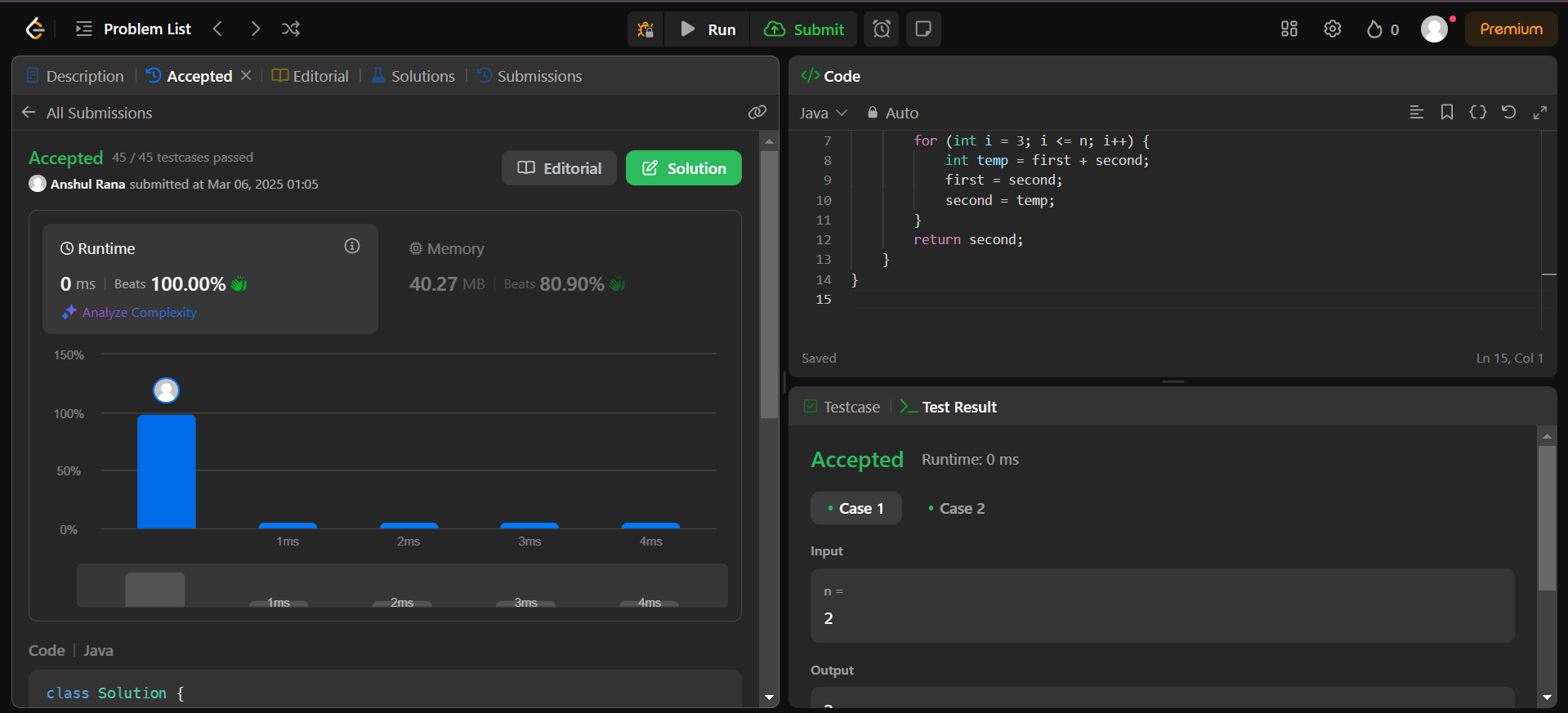
return n

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/1564063485/>

**Ques 2:**

**Aim:** Maximum Subarray

**Code:**

class Solution(object):

def maxSubArray(self, nums):

max\_sum = cur\_sum = nums[0]

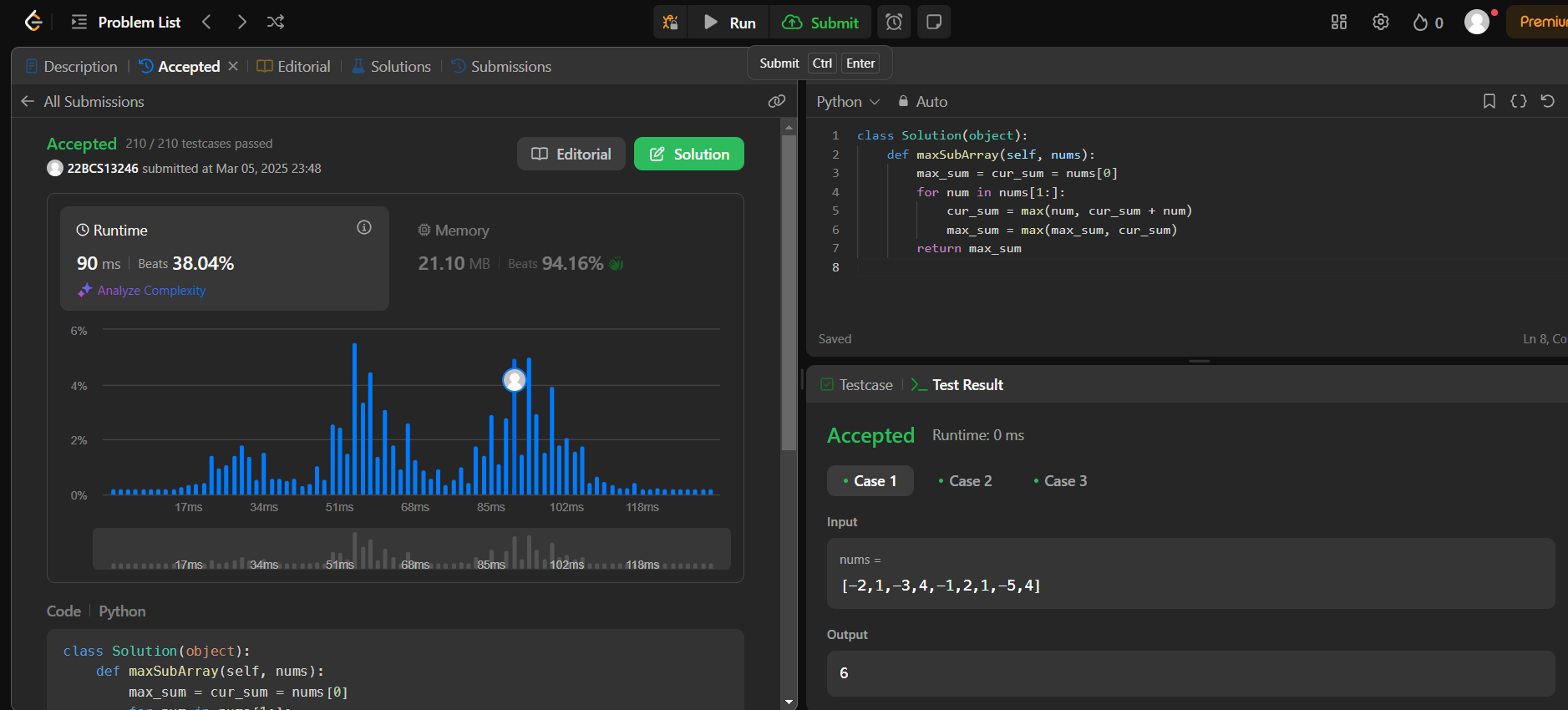
for num in nums[1:]:

cur\_sum = max(num, cur\_sum + num)

max\_sum = max(max\_sum, cur\_sum)

return max\_sum

**Submission Screenshot:**

****

**Submission Link:**

https://leetcode.com/problems/maximum-subarray/submissions/1564066747/

**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

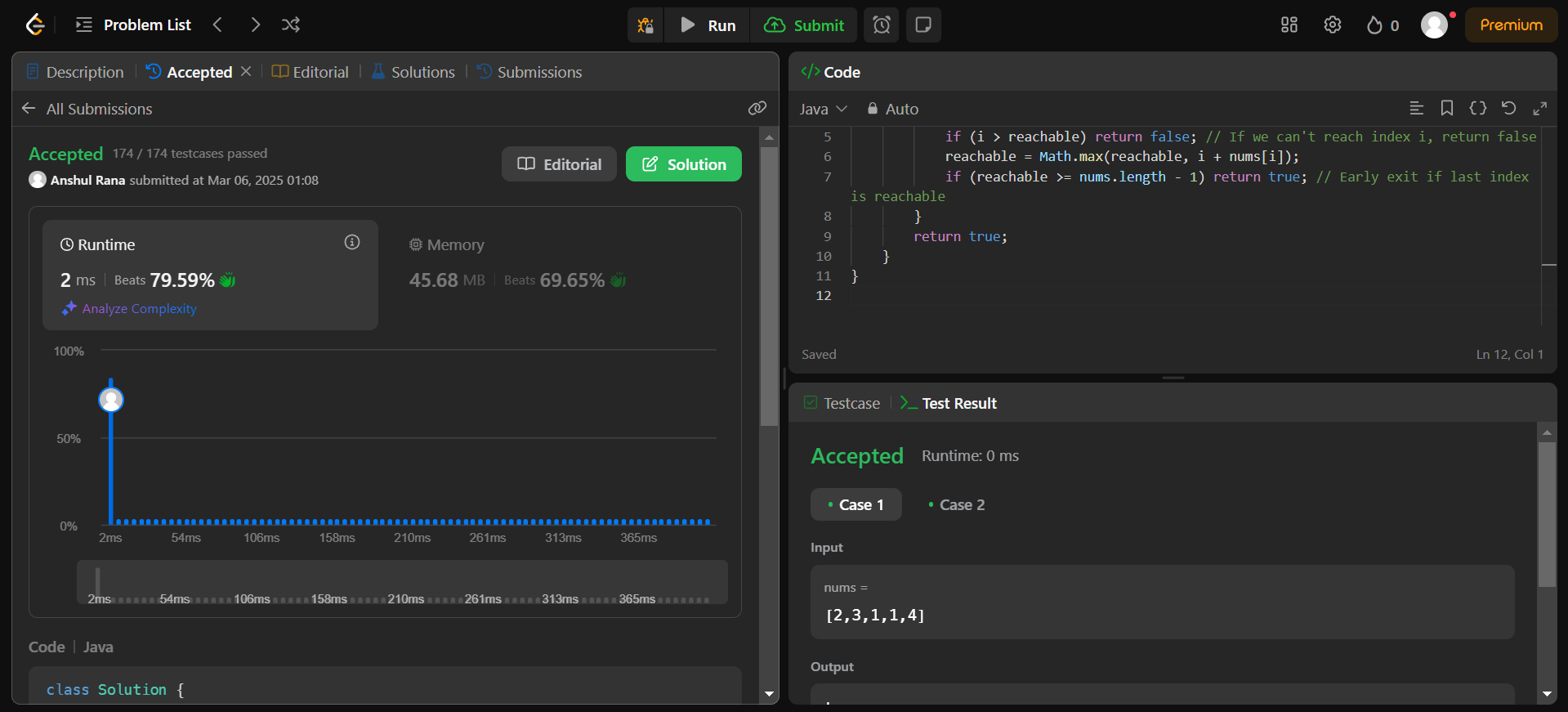
max\_reach = max(max\_reach, i + num)

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/1564068298/