

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**Subject Name:**

**Subject Code:** 20CSP351

**Submitted to: Submitted by:**

Er. Daulat Sihag (E13701) Name: Yash Gupta

UID: 20BCS5009

Section: 20BCS-DM-716

Group: B

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ex. No** | **List of Experiments** | **Conduct (MM: 12)** | **Viva**  **(MM: 10)** | **Record (MM: 8)** | **Total**  **(MM: 30)** | **Remarks/Signature** |
| 1.1 | Arrays,  Stacks, Queues linked list |  |  |  |  |  |
| 1.2 | String  Matching |  |  |  |  |  |
| 1.3 | Heap model |  |  |  |  |  |
| 2.1 | Hashing |  |  |  |  |  |
| 2.2 | Trees |  |  |  |  |  |
| 2.3 | Graph |  |  |  |  |  |
| 2.4 | Divide  and conquer |  |  |  |  |  |
| 3.1 | Greedy |  |  |  |  |  |
| 3.2 | Backtracking |  |  |  |  |  |
| 3.3 | Dynamic  Programming |  |  |  |  |  |

**Experiment 1.1**

**Student Name: Yash Gupta UID: 20BCS5009**

**Branch: BE-CSE Section/Group:20BCS\_DM-716 B**

**Semester: 6 Date of Performance:**

**Subject Name: CC LAB Subject Code: 20CSP\_351**

## Aim:

To implement the concept of arrays, queues, stacks and linked-list.

## Objective:

* + The objective is to build problem solving capability and to learn the basic concepts of data structures.
  + The implementation of Jump Game-II which shows and brushes up the concept of arrays.
  + The implementation of removing the duplicity in the Sorted-List-II.

## LeetCode code and output:

* **JUMP GAME-II**

class Solution:

def jump(self, nums: List[int]) -> int:

stk = []

mx,stp, endpt,i, cnt = 0, 0, 0, 0, 0

while i < len(nums)-1:

reach = i+nums[i]

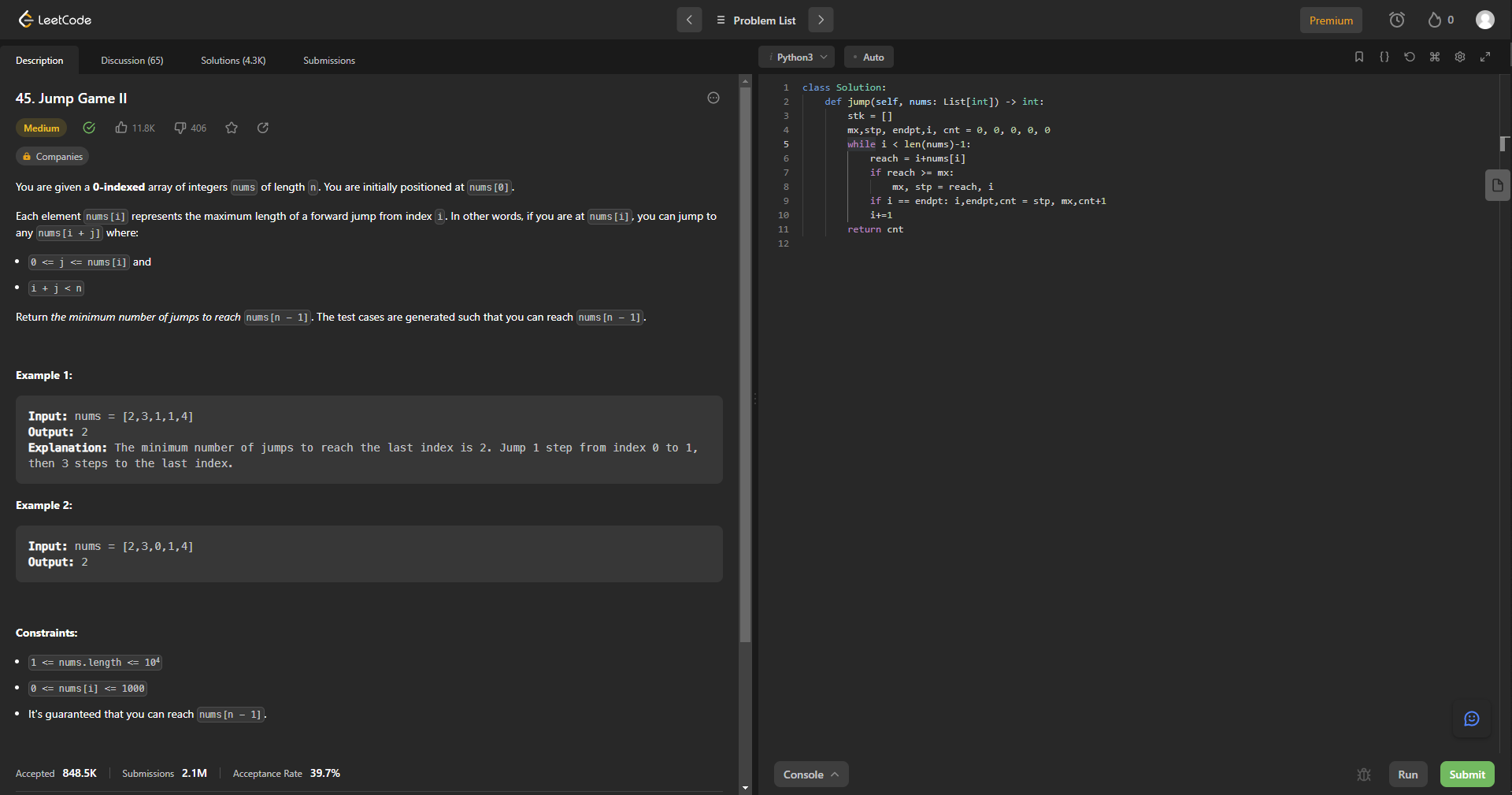
if reach >= mx:

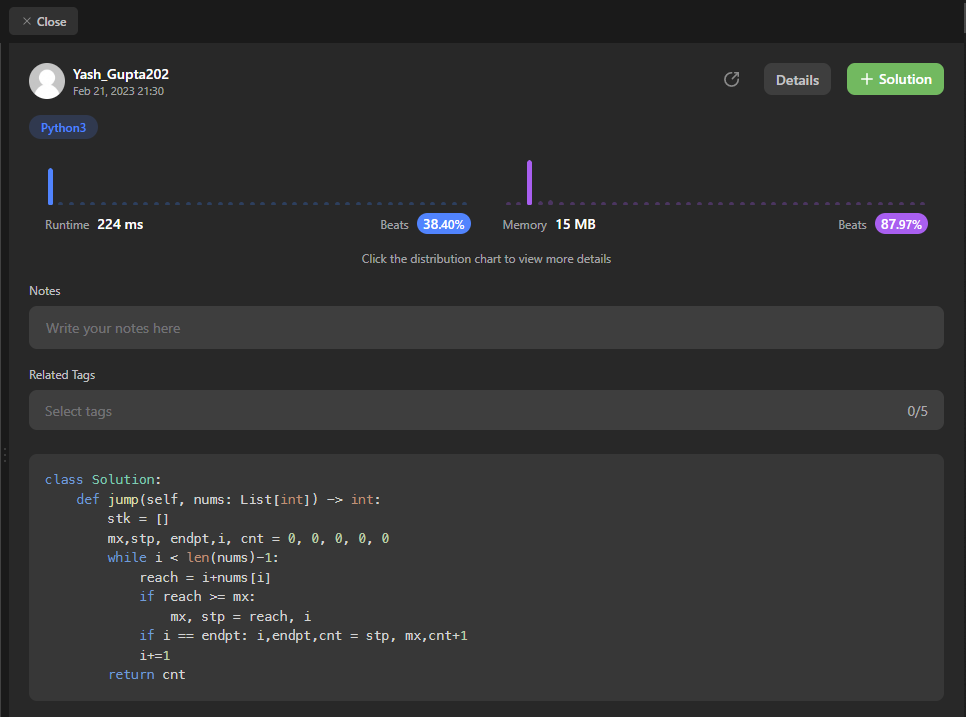
mx, stp = reach, i

if i == endpt: i,endpt,cnt = stp, mx,cnt+1

i+=1

return cnt





* **3Sum**

class Solution:

def threeSum(self, nums: List[int]) -> List[List[int]]:

res = []

nums.sort()

for i in range(len(nums)-2):

if i > 0 and nums[i] == nums[i-1]:

continue

l, r = i+1, len(nums)-1

while l < r:

s = nums[i] + nums[l] + nums[r]

if s < 0:

l +=1

elif s > 0:

r -= 1

else:

res.append((nums[i], nums[l], nums[r]))

while l < r and nums[l] == nums[l+1]:

l += 1

while l < r and nums[r] == nums[r-1]:

r -= 1

l += 1; r -= 1

return res

