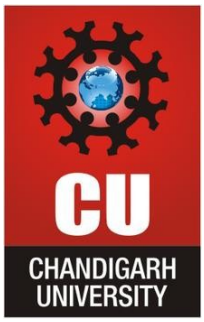
CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Submitted By: Vivek Kumar(21BCS8129)	**Submitted To:** Jayesh Surana(E13219)
Subject Name	Competitive Coding - II
Subject Code	20CSP-351
Branch	Computer Science and Engineering
Semester	6th

Experiment No. - 1

Student Name: Vivek Kumar
Branch: BE-CSE(LEET)
Semester: 6th
Subject Name: Competitive coding - II

UID: 21BCS8129
Section/Group: 20BCS-ST-801/B
Date of Performance: 14/02/2023
Subject Code: 20CSP-351

1. Aim/Overview of the practical:

Jump Game II

You are given a **0-indexed** array of integers `nums` of length `n`. You are initially positioned at `nums[0]`. Each element `nums[i]` represents the maximum length of a forward jump from index `i`. In other words, if you are at `nums[i]`, you can jump to any `nums[i + j]` where:

- `0 <= j <= nums[i]` and
- `i + j < n`

Return *the minimum number of jumps to reach* `nums[n - 1]`. The test cases are generated such that you can reach `nums[n - 1]`.

https://leetcode.com/problems/jump-game-ii/

2. Apparatus / Simulator Used:

- Windows 7 or above
- Google Chrome

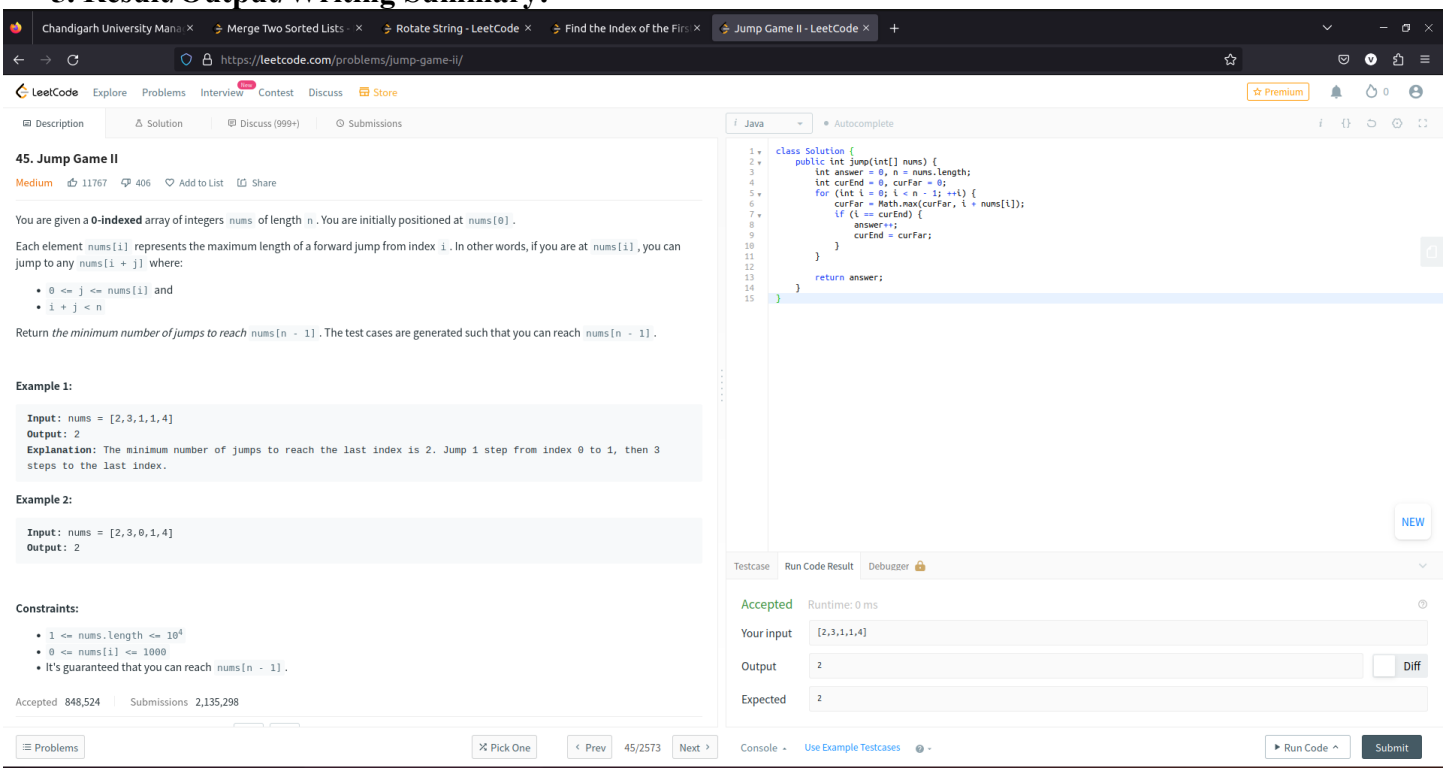
3. Objective:

- To understand the concept of Array and Jump Concept
- To implement the concept of Array Implementation.

4. Code:

```
class Solution {
    public int jump(int[] nums) {
        int answer = 0, n = nums.length;
        int curEnd = 0, curFar = 0;
        for (int i = 0; i < n - 1; ++i) {
            curFar = Math.max(curFar, i + nums[i]);
            if (i == curEnd) {
                answer++;
                curEnd = curFar;
            }
        }

        return answer;
    }
}
```

5. Result/Output/Writing Summary:



45. Jump Game II

Medium 11767 406 Add to List Share

You are given a **0-indexed** array of integers `nums` of length `n`. You are initially positioned at `nums[0]`.

Each element `nums[i]` represents the maximum length of a forward jump from index `i`. In other words, if you are at `nums[i]`, you can jump to any `nums[i + j]` where:

- `0 <= j <= nums[i]` and
- `i + j < n`

Return *the minimum number of jumps to reach* `nums[n - 1]`. The test cases are generated such that you can reach `nums[n - 1]`.

Example 1:

```
Input: nums = [2,3,1,1,4]
Output: 2
Explanation: The minimum number of jumps to reach the last index is 2. Jump 1 step from index 0 to 1, then 3 steps to the last index.
```

Example 2:

```
Input: nums = [2,3,0,1,4]
Output: 2
```

Constraints:

- `1 <= nums.length <= 10^4`
- `0 <= nums[i] <= 1000`
- It's guaranteed that you can reach `nums[n - 1]`.

Accepted 848,524 Submissions 2,135,298

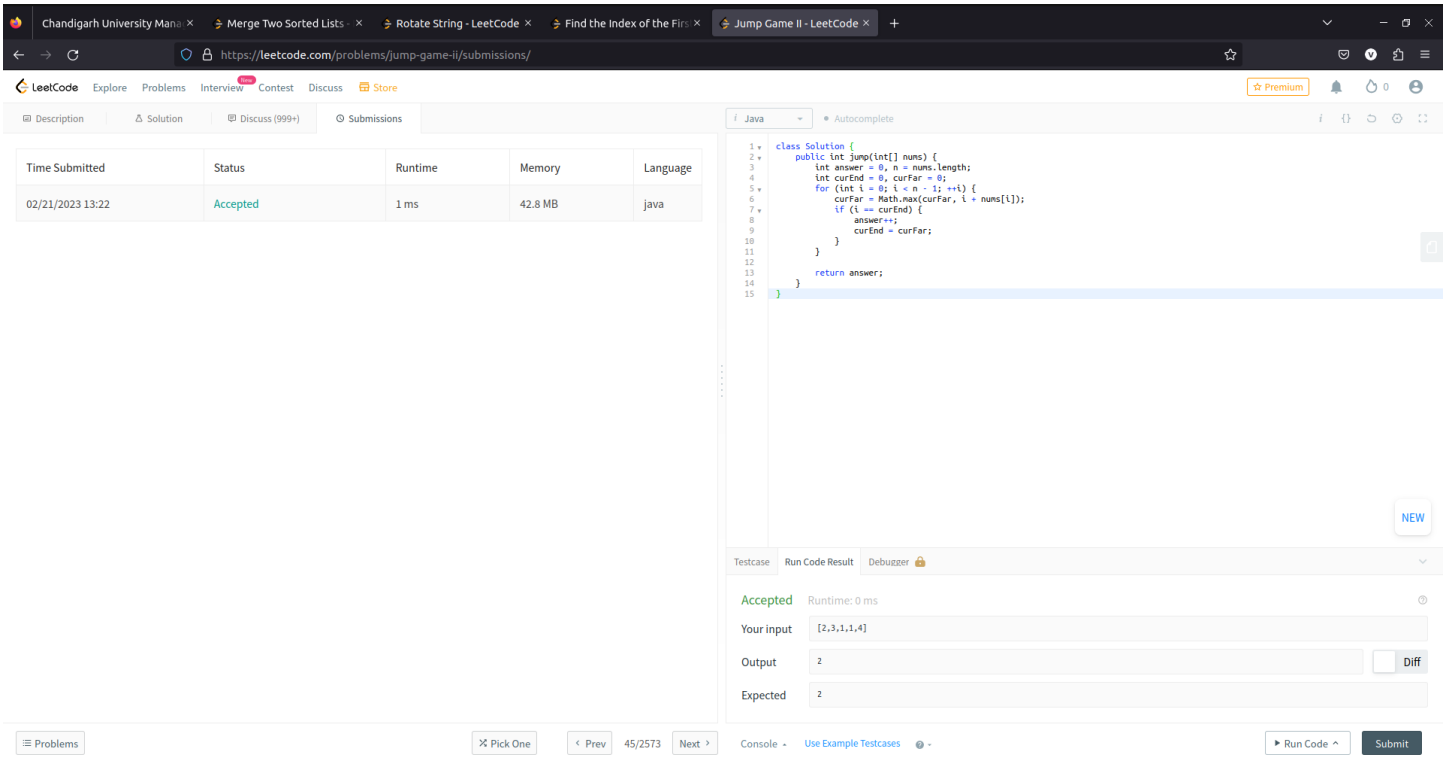
```
class Solution {
    public int jump(int[] nums) {
        int answer = 0, n = nums.length;
        int curEnd = 0, curFar = 0;
        for (int i = 0; i < n - 1; ++i) {
            curFar = Math.max(curFar, i + nums[i]);
            if (i == curEnd) {
                answer++;
                curEnd = curFar;
            }
        }

        return answer;
    }
}
```

Accepted Runtime: 0 ms

Your input [2,3,1,1,4]

Output 2

Expected 2



Time Submitted	Status	Runtime	Memory	Language
02/21/2023 13:22	Accepted	1 ms	42.8 MB	java

1. **Aim/Overview of the practical:**

   Merge Two Sorted List

   You are given the heads of two sorted linked lists `list1` and `list2`.
   Merge the two lists in a one **sorted** list. The list should be made by splicing together the nodes of the first two lists.
   Return *the head of the merged linked list*.
   https://leetcode.com/problems/merge-two-sorted-lists/

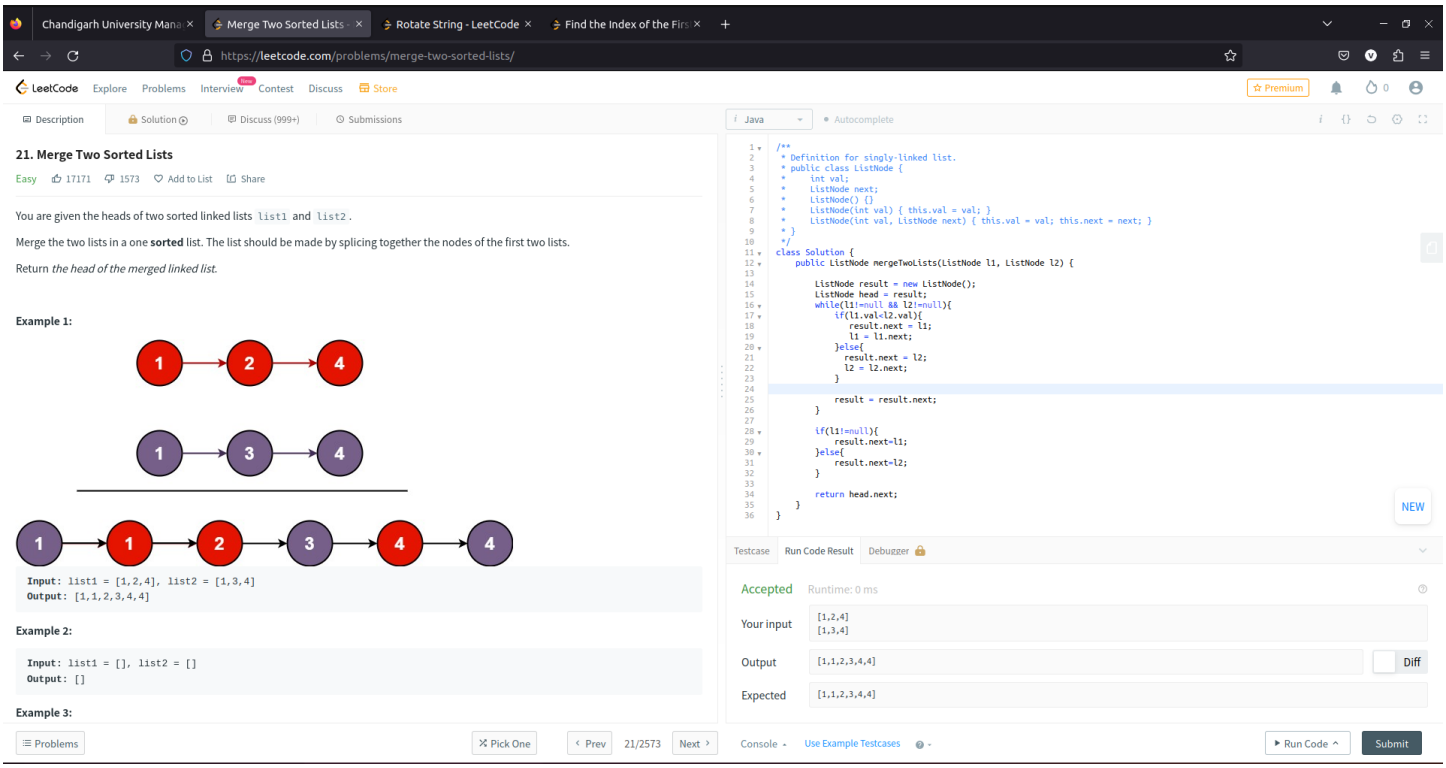
2. **Apparatus / Simulator Used:**
   - Windows 7 or above
   - Google Chrome

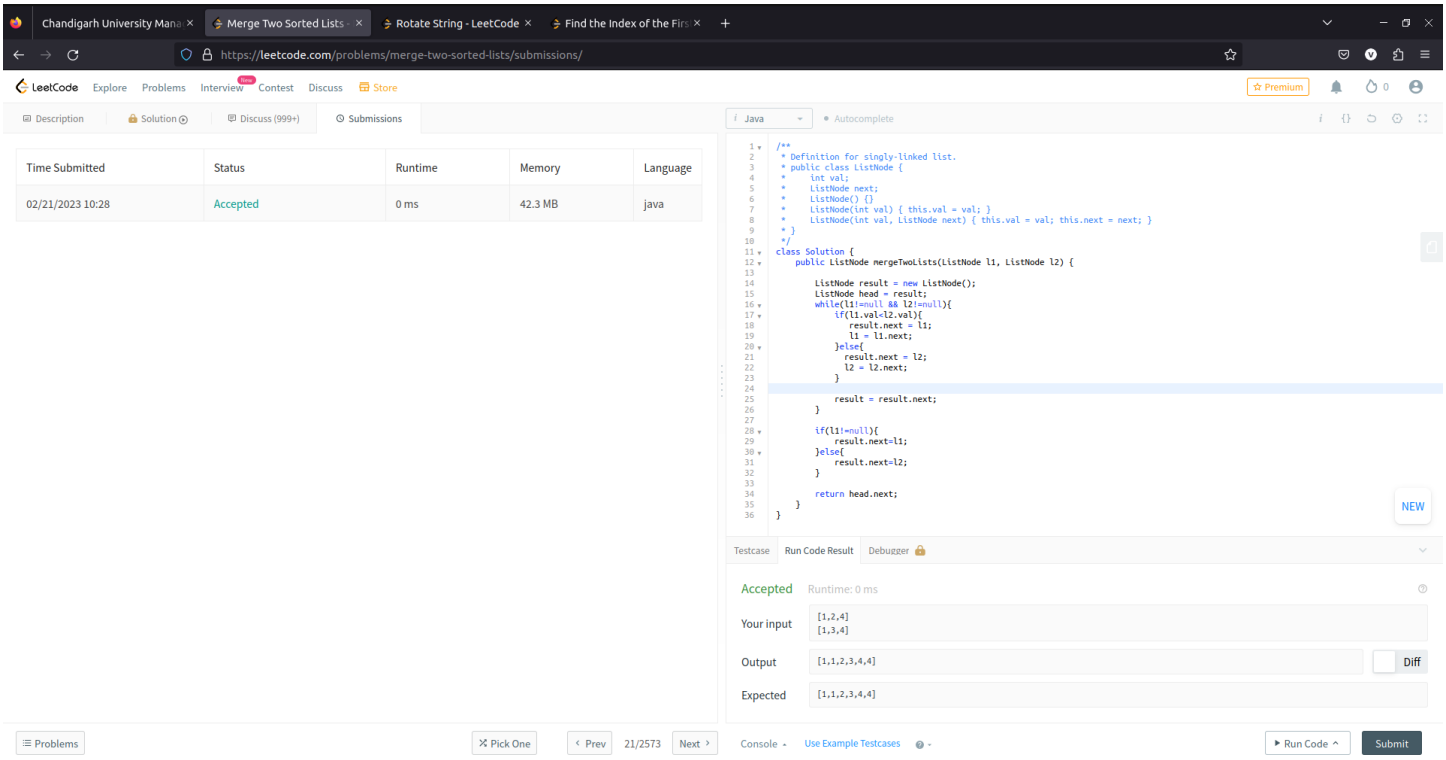
3. **Objective:**
   - To understand the concept of List and Node
   - To implement the concept of Sorting and Merge.

4. **Code:**

```
class Solution {
    public ListNode mergeTwoLists(ListNode l1, ListNode l2) {

        ListNode result = new ListNode();
        ListNode head = result;
        while(l1!=null && l2!=null){
            if(l1.val<l2.val){
               result.next = l1;
               l1 = l1.next;
            }else{
              result.next = l2;
              l2 = l2.next;
            }

            result = result.next;
        }

        if(l1!=null){
            result.next=l1;
        }else{
            result.next=l2;
        }

        return head.next;
    }
}
```

5. Result/Output/Writing Summary:





Learning outcomes (What I have learnt):

- Learned the concept of LinkedList.
- Learnt about Array in Merging And Sorting.

Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day).		
2.	Post-Lab Quiz Result.		
3.	Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions.		
	Signature of Faculty (with Date):	Total Marks Obtained:	