

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

# Subject Name: Competitive Coding 2

**Subject Code:** 20CSP-351

**Submitted to: Submitted by:**

Faculty name: Mr. Ankesh Gupta Name: Sahil Kaundal

UID: 21BCS8197

Section: 616

Group: A

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ex. No** | **List of Experiments** | **Conduct (MM: 12)** | **Viva**  **(MM: 10)** | **Record (MM: 8)** | **Total**  **(MM: 30)** | **Remarks/Signature** |
| 1. | Arrays, Stacks, Queues linked list |  |  |  |  |  |
| 2. |  |  |  |  |  |  |
| 3. |  |  |  |  |  |  |
| 4. |  |  |  |  |  |  |
| 5. |  |  |  |  |  |  |
| 6. |  |  |  |  |  |  |
| 7. |  |  |  |  |  |  |
| 8. |  |  |  |  |  |  |
| 9. |  |  |  |  |  |  |
| 10. |  |  |  |  |  |  |

**Experiment 1.1**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 6th **Date of Performance:** 22/02/2023

**Subject Name:** CC-2 Lab **Subject Code:** 20CSP-351

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

Implement 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]

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

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

**3. Code:**

class Solution {

public:

    int jump(vector<int>& N) {

        int len = N.size() - 1, curr = -1, next = 0, ans = 0;

        for (int i = 0; next < len; i++) {

            if (i > curr) ans++, curr = next;

            next = max(next, N[i] + i);

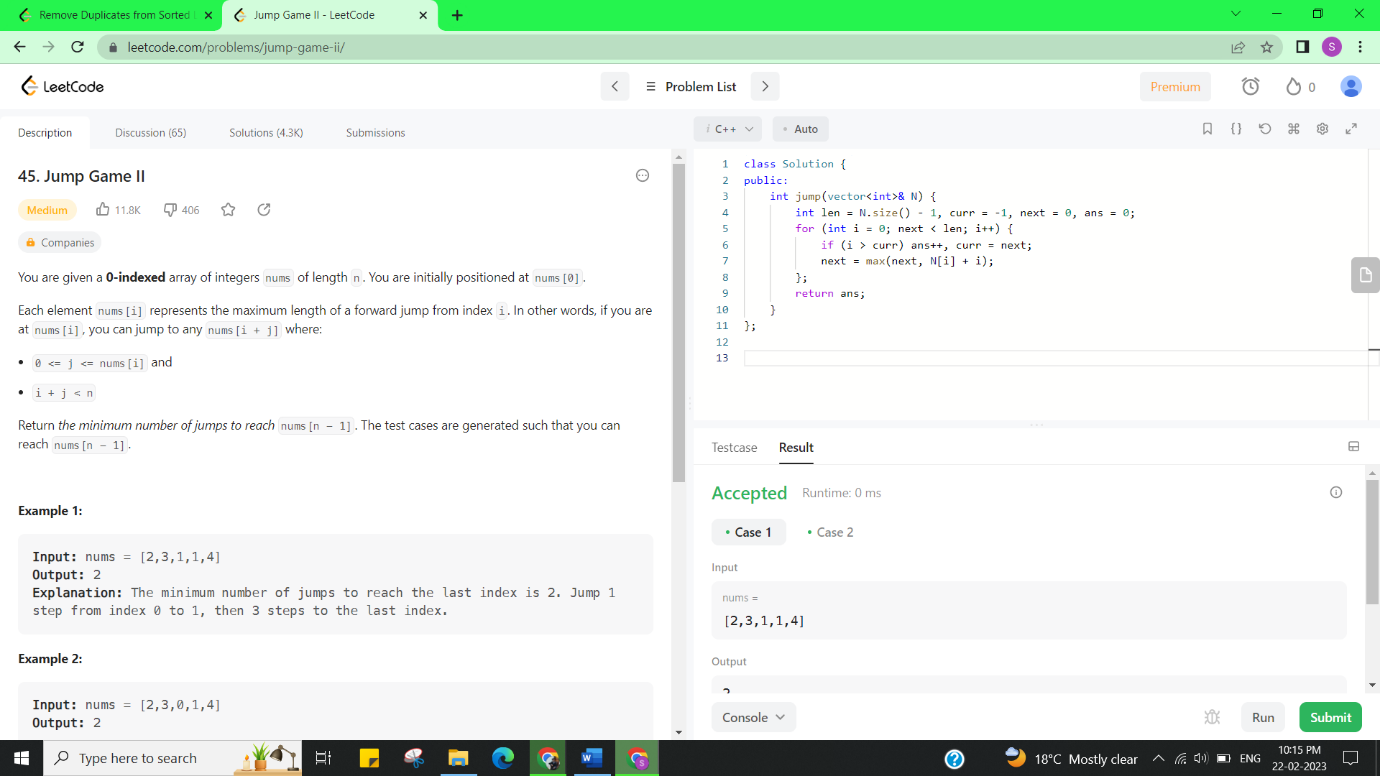
        };

        return ans;

    }

};

**4. Result/Output/Writing Summary:**



**Experiment 1.2**

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

Remove the duplicate elements from list.

Given the head of a sorted linked list, delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list. Return the linked list ***sorted*** as well.

<https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii/>

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Code:**

class Solution {

public:

    ListNode\* deleteDuplicates(ListNode\* head) {

        ListNode\* dummy = new ListNode(0);

        dummy->next = head;

        ListNode\* cur = dummy;

        int duplicate;

        while (cur->next && cur->next->next) {

            if (cur->next->val == cur->next->next->val) {

                duplicate = cur->next->val;

                while (cur->next && cur->next->val == duplicate) {

                    cur->next = cur->next->next;

                }

            }

            else {

                cur = cur->next;

            }

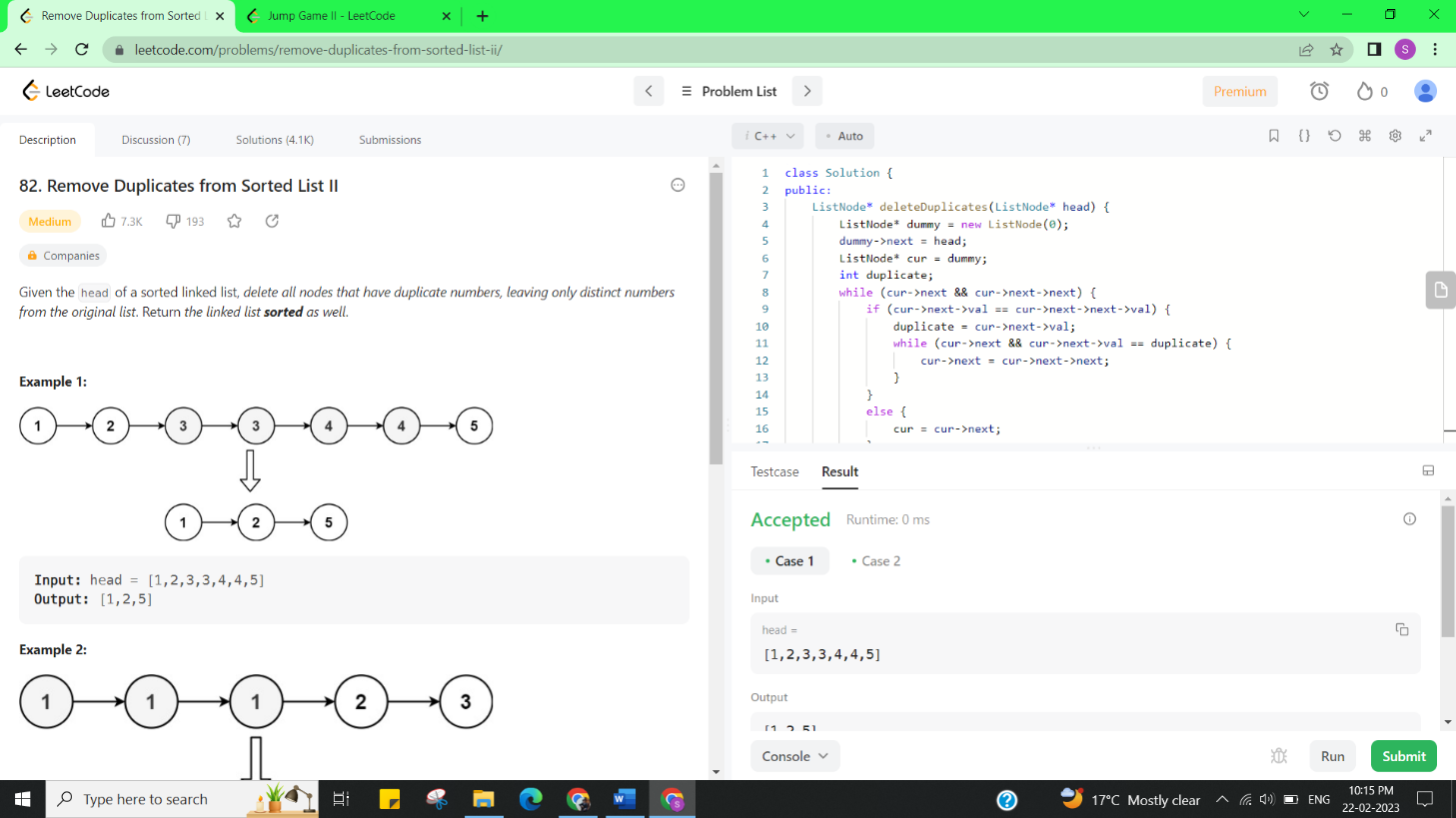
        }

        return dummy->next;

    }

};

1. **Result/Output/Writing Summary:**

****

**Learning outcomes (What I have learnt):**

* + Learned the concept of jump game-2.
  + Learnt about Remove the duplicate elements from list.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |