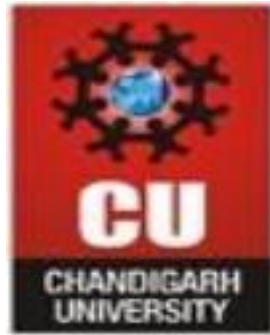


# **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



## **UNIVERSITY INSTITUTE OF ENGINEERING** **Department of Computer Science & Engineering**

**Subject Name:**

**Subject Code: 20CSP351**

**Submitted to:**

Er. Vipasha Sharma

**Submitted by:**

Name: Kanishk Soni

UID: 20BCS9398

Section: 20BCS\_DM\_708

Group: B

# **DEPARTMENT OF**

# **COMPUTER SCIENCE & ENGINEERING**

## **INDEX**

<b>Ex. No</b>	<b>List of Experiments</b>	<b>Conduct (MM: 12)</b>	<b>Viva (MM: 10)</b>	<b>Record (MM: 8)</b>	<b>Total (MM: 30)</b>	<b>Remarks/Signature</b>
1.1	Arrays, Queues, Stack and Linked List					
1.2						
1.3						
2.1						
2.2						
2.3						
2.4						
3.1						
3.2						
3.3						

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## Experiment-1.1

**Student Name:** Kanishk Soni

**Branch:** BE-CSE

**Semester:** 6<sup>th</sup>

**Subject Name:** Competitive Coding-II

**UID:** 20BCS9398

**Section/Group:** 20BCS-DM\_708B

**Date of Performance:** 23-02-2023

**Subject Code:** 20CSP-351

**AIM:** To implement the concept of Arrays, Queues, Stack and Linked List

**Problem1:** Implement Jump Game

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

**Program Code:**

```
class Solution {  
    public int jump(int[] A) {  
        int jumps = 0, curEnd = 0, curFarthest = 0;  
        for (int i = 0; i < A.length - 1; i++) {  
            curFarthest = Math.max(curFarthest, i + A[i]);  
            if (i == curEnd) {  
                jumps++;  
                curEnd = curFarthest;  
            }  
        }  
        return jumps;  
    }  
}
```

**Output:**

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LeetCode

Problem List

Premium

Description Discussion (65) Solutions (4.3K) Submissions

Accepted

Next question

46. Permutations

More challenges

55. Jump Game 1306. Jump Game III 1871. Jump Game VII

All statuses All languages

Accepted a few seconds ago Java

Accepted Feb 08, 2023 Java

Accepted Jan 26, 2023 Java

Close

taabish19 Feb 22, 2023 10:43

Java

Runtime 2 ms Beats 40.8% Memory 49.1 MB Beats 5.6%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags 0/5

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

Console Run Submit

Problem2: Remove the duplicate elements from list

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

Program Code:

```
class Solution {
    public ListNode deleteDuplicates(ListNode head) {
        if (head == null) return null;

        ListNode ans = new ListNode();
        ans.next = head;
        ListNode ptr = ans;

        while (ptr.next != null && ptr.next.next != null) {

            if (ptr.next.val == ptr.next.next.val) {
                int value = ptr.next.val;
                while (ptr.next != null && ptr.next.val == value) {
                    ptr.next = ptr.next.next;
                }
            } else {

```

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
        ptr = ptr.next;
    }
}
return ans.next;
}
```

Output:

The screenshot shows a LeetCode submission interface. The top navigation bar includes the LeetCode logo, a 'Problem List' menu, and user status (Premium, 0 problems solved). The main content area is divided into two panels. The left panel shows the submission status as 'Accepted' with a green checkmark, and lists 'Next question' and 'More challenges'. The right panel displays the submission details for user 'taabish19' on Feb 22, 2023, at 10:57. It shows the problem '83. Remove Duplicates from Sorted List' with a 'Java' language tag. Performance metrics include Runtime (1 ms), Beats (25.82%), Memory (43.6 MB), and Beats (10.44%). A distribution chart is visible. Below the metrics, there is a 'Notes' section with a placeholder 'Write your notes here', a 'Related Tags' section with a 'Select tags' button, and a code editor showing the solution code. The code is as follows:

```
class Solution {
    public ListNode deleteDuplicates(ListNode head) {
        if (head == null) return null;

        ListNode ans = new ListNode();
        ans.next = head;

        ListNode ptr = ans;
```

At the bottom right, there is a 'Console' section with a 'Run' button and a 'Submit' button.