

DSA Assignment - 19 June

Name - Arpit Mahala

Roll.No - 23/11/EC/025

LeetCode Id - <https://leetcode.com/u/Arpit-Mahala/>

GeekforGeeks- <https://www.geeksforgeeks.org/user/arpitmay36m/>

Github Repo Link -

https://github.com/Arpit-Mahala1/2025_Internship_Training

1. Find the Town Judge

○ Platform: LeetCode

○ Link: <https://leetcode.com/problems/find-the-town-judge/>

The screenshot shows a LeetCode submission interface for the 'Find the Town Judge' problem. The submission is in C++ and has been accepted. The left sidebar displays the submission status: 'Accepted' with 92/92 testcases passed, submitted by 'Arpit Mahala' on Jun 20, 2025 at 21:17. It also shows performance metrics: Runtime 0 ms (Beats 100.00%) and Memory 64.91 MB (Beats 93.84%). A bar chart shows the runtime distribution. The main area displays the C++ code for the 'findJudge' function. The right sidebar shows the 'Testcase' tab with 'Case 1' selected, displaying the input 'n = 2' and 'trust = '.

Problem List < > < >

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 92 / 92 testcases passed

Arpit Mahala submitted at Jun 20, 2025 21:17

Editorial Solution

Runtime 0 ms | Beats 100.00% | Memory 64.91 MB | Beats 93.84%

Analyze Complexity

40% 20% 0%

0ms 10ms 20ms 30ms 40ms 50ms 60ms

Code | C++

```
class Solution {
public:
    int findJudge(int n, vector<vector<int>>& trust) {
        vector<int> trustScore(n + 1, 0);

        for(int i = 0; i < trust.size(); i++) {
            int a = trust[i][0];
            int b = trust[i][1];

            trustScore[a]--;
            trustScore[b]++;
        }

        for(int i = 1; i <= n; i++) {
            if(trustScore[i] == n - 1) {
                return i;
            }
        }

        return -1;
    }
};
```

Saved Ln 4, Col 42

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

n = 2

trust =

2. Plus One

- Platform: LeetCode
- Link: <https://leetcode.com/problems/plus-one/description/>

The screenshot displays the LeetCode submission interface for the 'Plus One' problem. The top navigation bar includes 'Description', 'Accepted', 'Editorial', 'Solutions', and 'Submissions'. The submission status is 'Accepted' with 111/111 testcases passed, submitted by 'Arpit Mahala' on Jun 20, 2025 at 22:05. The 'Solution' tab is active, showing a C++ code editor with the following code:

```
1 class Solution {
2 public:
3     vector<int> plusOne(vector<int>& digits) {
4         int n = digits.size();
5         int carry = 1;
6         vector<int> result;
7
8         for (int i = n - 1; i >= 0; i--) {
9             int sum = digits[i] + carry;
10            result.push_back(sum % 10);
11            carry = sum / 10;
12        }
13
14        if (carry) {
15            result.push_back(carry);
16        }
17
18        vector<int> finalResult(result.size());
19        for (int i = result.size() - 1; i >= 0; i--) {
20            finalResult[i] = result[i];
21        }
22        return finalResult;
23    }
24 }
```

Below the code editor, the 'Testcase' and 'Test Result' panels are visible. The 'Test Result' panel shows 'Accepted' with a runtime of 0 ms. The input for Case 1 is 'digits = [1, 2, 3]', and the output is '[1, 2, 3]'. A runtime performance graph is also shown, indicating a runtime of 0 ms and a memory usage of 11.64 MB, which beats 5.67% of other submissions.

3. Maximum Subarray

- Platform: LeetCode

- Link:

<https://leetcode.com/problems/maximum-subarray/description/>

The screenshot displays a LeetCode submission interface for the 'Maximum Subarray' problem. The submission is marked as 'Accepted' with 210/210 testcases passed, submitted by 'Arpit Mahala' on Jun 21, 2025 at 23:11. The runtime is 0 ms, beating 100.00% of submissions, and the memory usage is 71.74 MB, beating 52.12%. A bar chart shows the submission's performance relative to others. The code is written in C++ and implements a Kadane's algorithm. The test case input is [-2,1,-3,4,-1,2,1,-5,4].

Runtime: 0 ms | Beats 100.00% | Memory: 71.74 MB | Beats 52.12%

Code:

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0];
        int currSum = nums[0];

        for (int i = 1; i < nums.size(); i++) {
            currSum = max(nums[i], currSum + nums[i]);
            maxSum = max(maxSum, currSum);
        }

        return maxSum;
    }
};
```

Testcase:

Case 1: nums = [-2,1,-3,4,-1,2,1,-5,4]

4. Assign Cookies

- Platform: LeetCode
- Link: <https://leetcode.com/problems/assign-cookies/description/>

