Experiment 1.1

Student Name: Avinash Kumar **UID: 21BCS8908**
Branch: BE-CSE **Section/Group: 21BCS_CC_648_B**
Semester: 6 **Date of Performance:16-01-2024**
Subject Name: Advance Programming lab **Subject Code: 21CSP-251**

1. Aim:

- To Solve the Jump Game II
- To Solve the 3 SUM Problem

To Solve the Jump Game II

Objective :

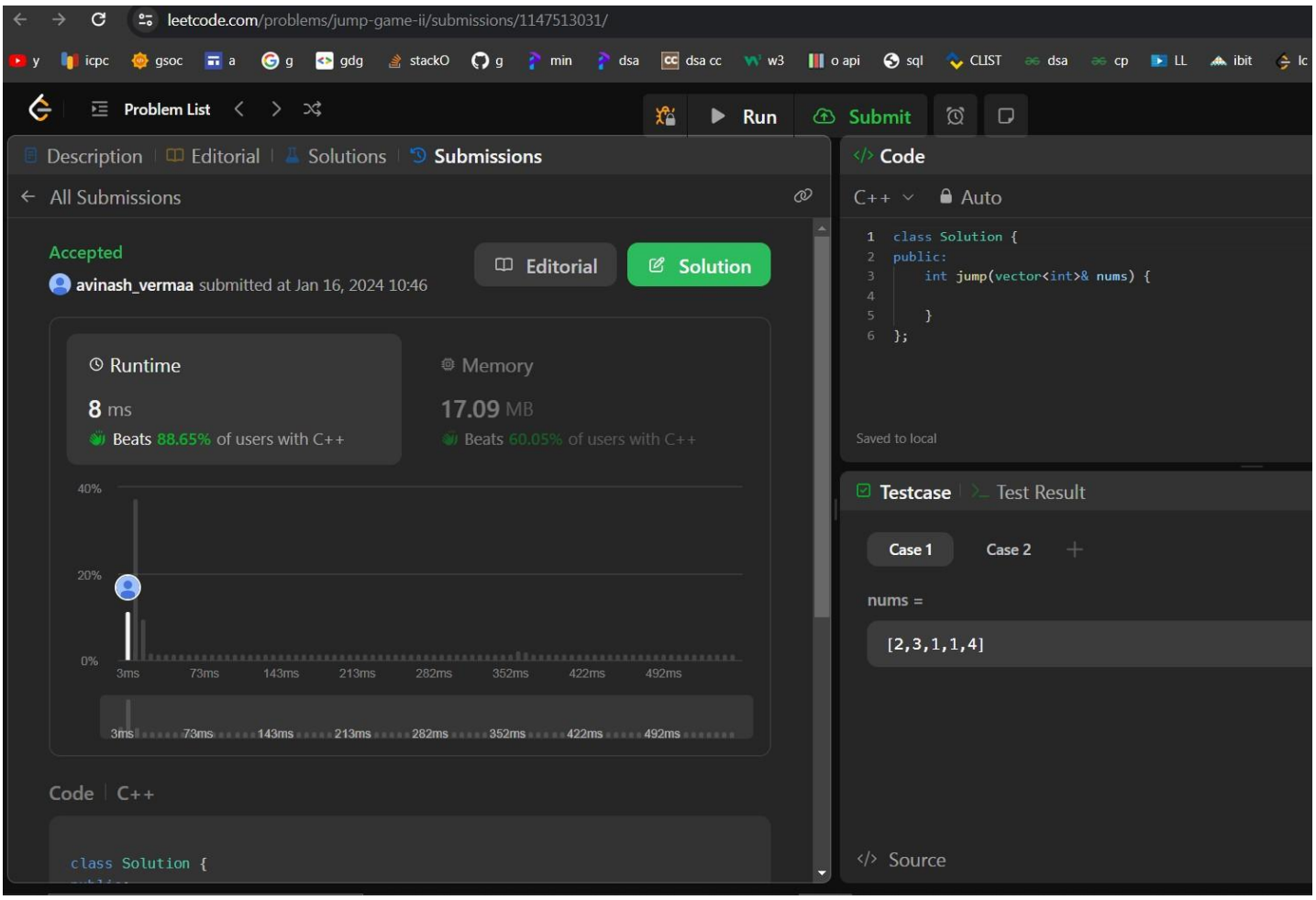
- You are given a 0-indexed array of integers nums of length n. You are initially positioned at nums[0].
- Each element nums[i] represents the maximum length of a forward jump from index i. In other words, if you are at nums[i], you can jump to any nums[i + j] where:
- 0 <= j <= nums[i] and
- i + j < 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]Givenanintegerarraynums,returnallthetriplets[nums[i],nums[j],nums[k]]such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.

2. Algo./Approach:

```
class Solution {
public:

    int jump(vector<int>& nums) {

      for(int i = 1; i < nums.size(); i++)
      {
```

```
        nums[i] = max(nums[i] + i, nums[i-1]);
      }
     long long int index = 0,ans = 0;

      while(index < nums.size() - 1)
      {
        ans++;
        index = nums[index];
      }

      return ans;
    }
};
```

OUTPUT 1:



Avinash Kumar
21BCS8908

To Solve the 3 SUM Problem

Objective :

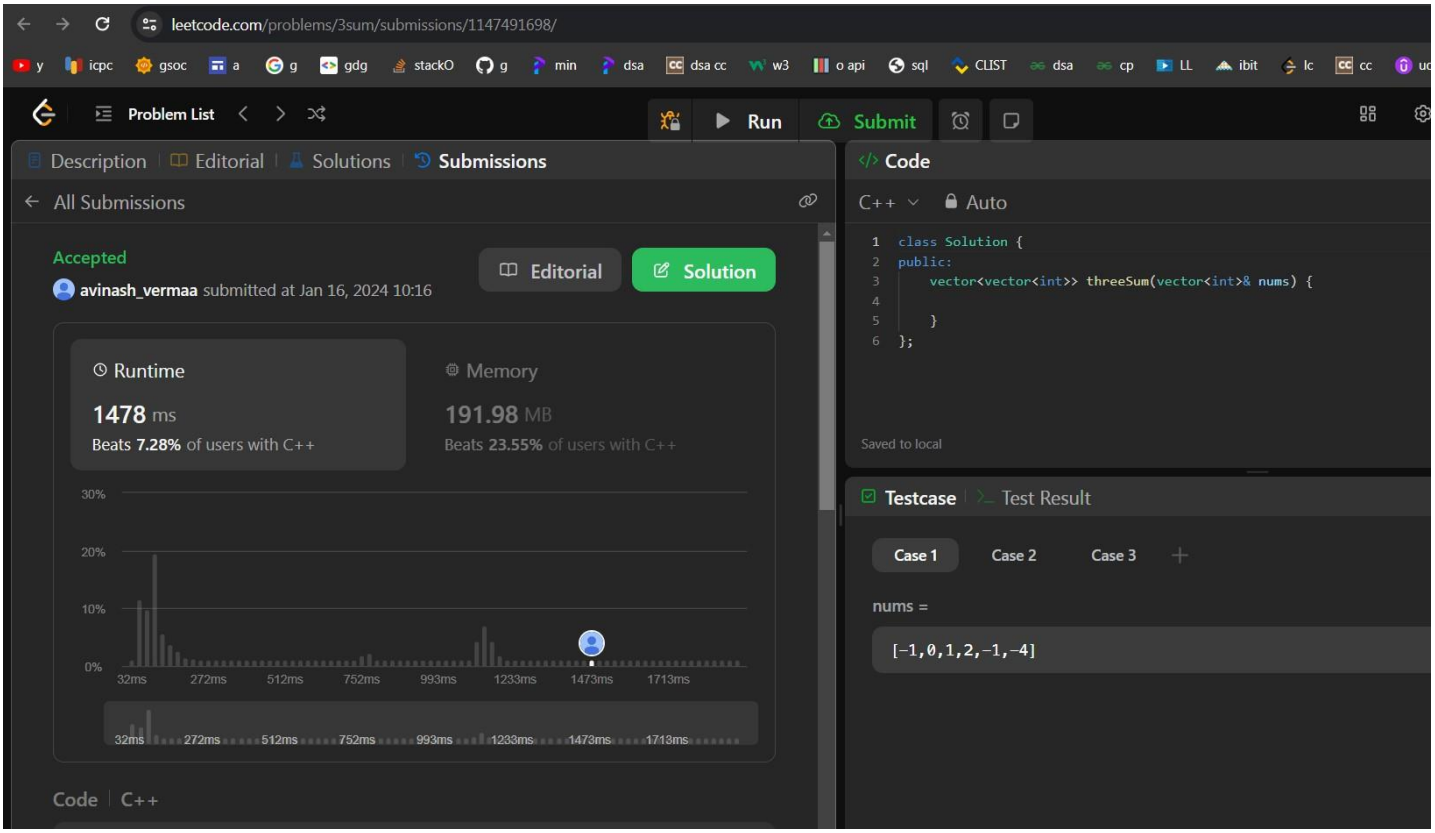
- Given an integer array nums,return all the triplets[nums[i],nums[j],nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.

Algo./Approach:

```
class Solution {
public:
    vector<vector<int>> threeSum(vector<int>& nums) {
        int target = 0;
        sort(nums.begin(), nums.end());
        set<vector<int>> s;
        vector<vector<int>> output;
        for (int i = 0; i < nums.size(); i++){
            int j = i + 1;
            int k = nums.size() - 1;
            while (j < k) {
                int sum = nums[i] + nums[j] + nums[k];
                if (sum == target) {
                    s.insert({nums[i], nums[j], nums[k]});
                    j++;
                    k--;
                }
                else if (sum < target) {
                    j++;
                }
                else {
                    k--;
                }
            }
        }
        for(auto triplets : s)
            output.push_back(triplets);
        return output;
    }
};
```

Avinash Kumar
21BCS8908

OUTPUT 2:



3. Learning Outcomes:

- Learnt the concept of Arrays and use of indexing in Arrays.
- Learnt the concept of Conditionals and loops and how to apply the min problem solving.
- Learnt the basic concept of problem solving and competitive programming.