CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Experiment-1.1

Student Name: Uday Pratap Singh
UID: 21BCS7439
Branch: BE-CSE
Section/Group:21BCSCC-641
Semester: 6th
Date of Performance:13-01-2024
Subject Name: Advanced Programming lab-2
Subject Code:21CSP-351

Aim:

- To Solve the 3 SUM Problem
- To Solve the merge two sorted linked lists

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.
- Trapping rainwater problem: Find the maximum amount of water that can be trapped within a given set of bars where each bar's width is 1 unit.

Algorithm:

3sum algorithm:

1. Sort the input array nums in non-decreasing order.
2. Initialize an empty vector (or list) to store the triplets that satisfy the given condition.
3. Iterate through the array nums using a variable i, starting from the first element (i = 0) and ending before the last element (i < nums. Size () - 2).
4. For the current element at index I, initialize two pointers j and k, pointing to the next element (i + 1) and the last element (nums. size () - 1) respectively.
5. While j is less than k, perform the following steps: a. Calculate the sum of the three elements at indices i, j, and k. b. If the sum is equal to 0, add the triplet [nums[i], nums[j], nums[k]] to the result vector and move the pointers j and k to the next and previous elements respectively, while skipping any duplicates.
6. If the sum is less than 0, move the pointer j to the next element. d. If the sum exceeds 0, move the pointer k to the previous element.

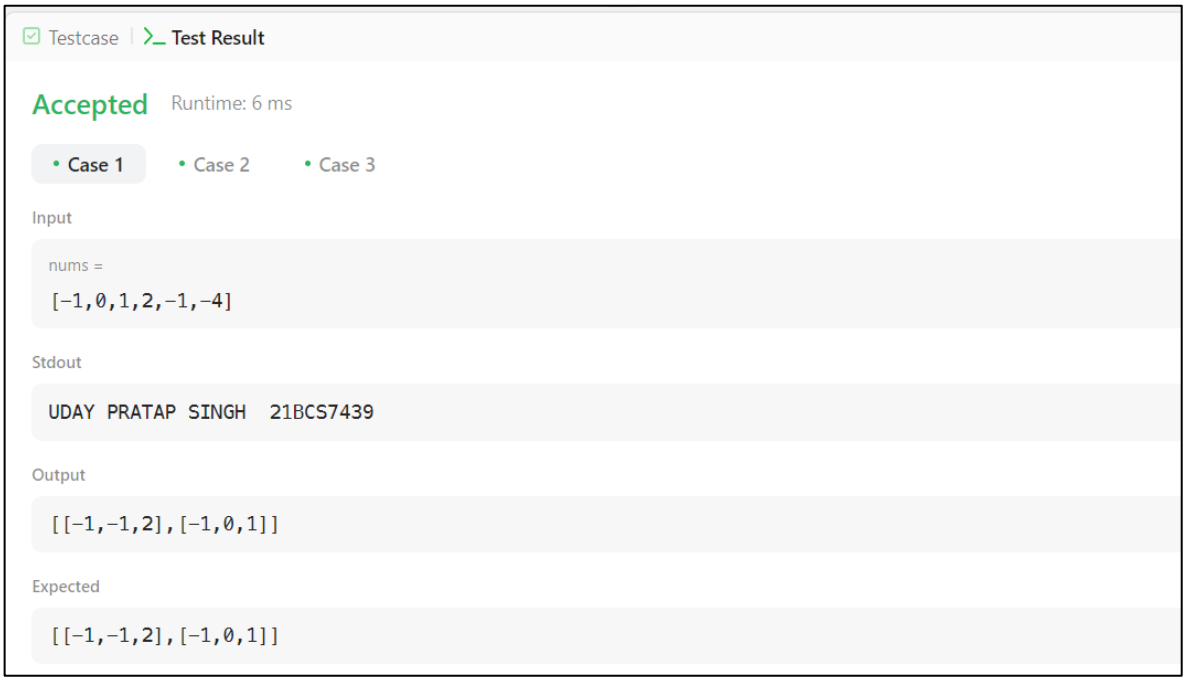
Tapping RainWater:

1. leftmax is initialized to the height of the first element in the array (height[0]).
2. rightmax is initialized to the height of the last element in the array (height[height.size()-1]).
3. leftpos is initialized to 1, representing the second element in the array.
4. rightpos is initialized to height.size()-2, representing the second-to-last element in the array.
5. answer is initialized to 0, which will be used to accumulate the trapped rainwater.

Code(A):

```
class Solution {
public:
    vector<vector<int>> threeSum(vector<int>& nums) {
        sort(nums.begin(),nums.end());

        cout<<"UDAY PRATAP SINGH"<<"  "<<"21BCS7439";

        vector<vector<int>>res;
        for(int i=0;i<nums.size()-2;i++)
        {
            if(i==0 || (i>0 && nums[i] != nums[i-1]))
            {
                int low=i+1;
                int hi=nums.size()-1;
                int sum=0-nums[i];
                while(low<hi)
                {
                    if(nums[low]+nums[hi]==sum)
                    {
                        vector<int>temp;
                        temp.push_back(nums[i]);
                        temp.push_back(nums[low]);
                        temp.push_back(nums[hi]);
                        res.push_back(temp);

                        while(low<hi && nums[low]==nums[low+1]) low++;
                        while(low<hi && nums[hi]==nums[hi-1]) hi--;

                        low++; hi--;
                    }
                    else if(nums[low]+nums[hi]<sum) low++;
                    else hi--;
                }
            }
        }
        return res;
    }
};
```

Output(A):



Code(B):

```
class Solution {
public:
    int trap(vector<int>& height) {

        cout<<"UDAY PRATAP SINGH"<<"  "<<"21BCS7439";

        int leftmax=height[0];
        int rightmax=height[height.size()-1];
        int leftpos=1;
        int rightpos=height.size()-2;
        int answer = 0;
        while(leftpos<=rightpos){
            if(height[leftpos]>=leftmax){
                leftmax=height[leftpos];
                leftpos++;
            }
            else if(height[rightpos]>=rightmax){
```

```
                rightmax=height[rightpos];
                rightpos--;
            }
            else if(leftmax<=rightmax && height[leftpos]<leftmax){
                answer+=leftmax-height[leftpos];
                leftpos++;
            }
            else{
                answer+=rightmax-height[rightpos];
                rightpos--;
            }
        }
        return answer;
    }
};
```

Output(B):