Experiment 1.1

Student Name: Rohan Jaiswal | **UID: 21BCS2856**
Branch: CSE | **Section/Group: KRG-CC1-B**
Semester: 6 | **Date of Performance: 16th jan**
Subject Name: Advanced Programming | **Subject Code: 21CSP-351**

Problem 1: 3Sum

15. 3Sum

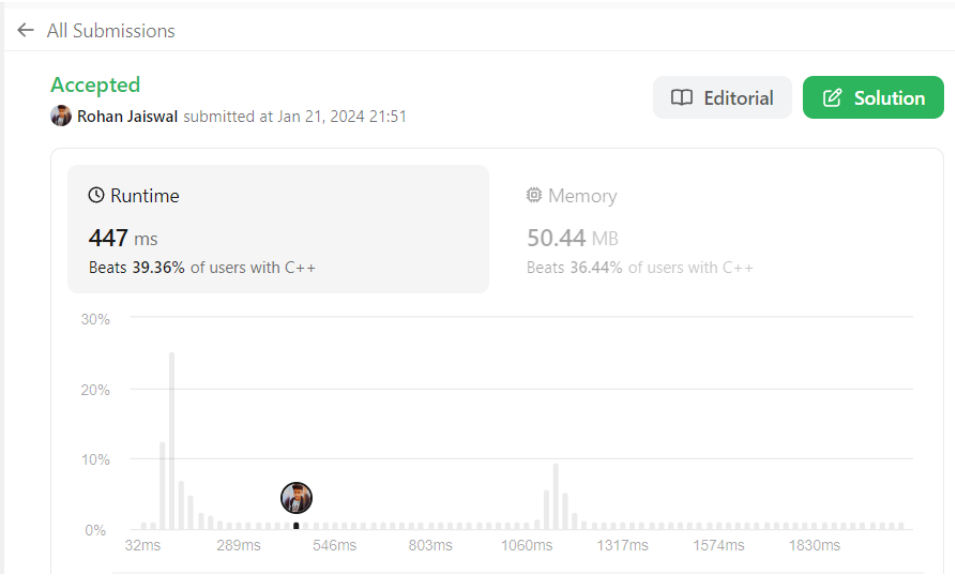
Solved

Medium | Topics | Companies | Hint

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`.

Notice that the solution set must not contain duplicate triplets.

Code and output:



3sum.cpp

```cpp
class Solution {
public:
    vector<vector<int>> threeSum(vector<int>& vec) {
        map<int, int> mp;
        for (auto i : vec)
            mp[i]++;
        vector<int> nums;
        for (auto p : mp) {
            for (int i = 0; i < min(3, p.second); i++)
                nums.push_back(p.first);
        }
        int n = nums.size();
        set<vector<int>> ans;
        for (int i = 0; i < n - 2; ++i) {
            int target = nums[i];
            for (int j = i + 1; j < n - 1; ++j) {
                int req = -1 * (target + nums[j]);
                auto lower = lower_bound(nums.begin() + j + 1, nums.end(), req);
                int idx = lower - nums.begin();
                if (lower != nums.end() && nums[idx] == req) {
                    vector<int> tmp;
                    tmp.push_back(nums[i]);
                    tmp.push_back(nums[j]);
                    tmp.push_back(req);
                    ans.insert(tmp);
                }
            }
        }
        vector<vector<int>> res;
        for (auto v : ans)
            res.push_back(v);
        return res;
    }
};
```

Problem 2: Merge Two Sorted Lists

21. Merge Two Sorted Lists

Solved

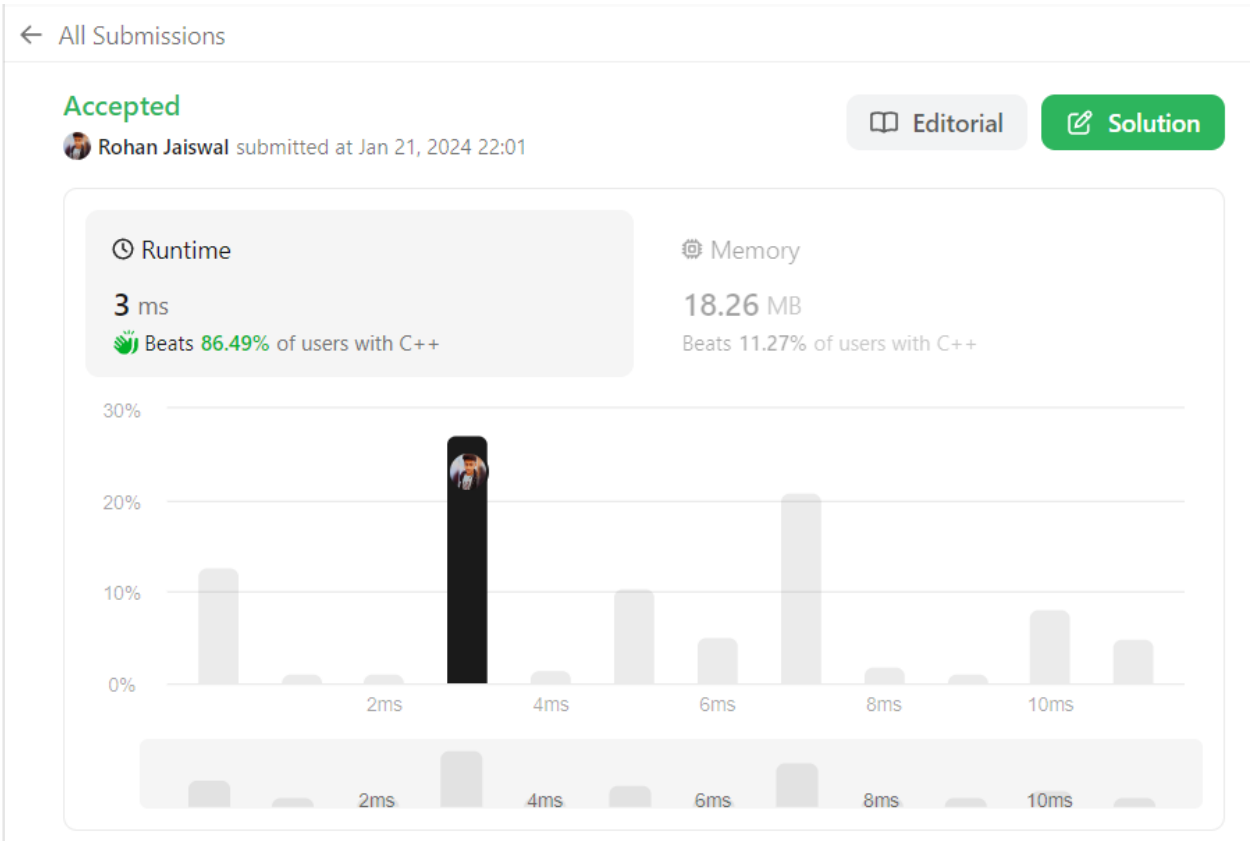
Easy | Topics | Companies

You are given the heads of two sorted linked lists `list1` and `list2`.

Merge the two lists into one **sorted** list. The list should be made by splicing together the nodes of the first two lists.

Return *the head of the merged linked list*.

Code and output:



mergeTwoLists.cpp

```cpp
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        if(list1==NULL || list2==NULL){
            return list1? list1 : list2;
        }
        ListNode* head;
        if(list1->val<list2->val){
            head = list1;
            list1=list1->next;
        }
        else{
            head = list2;
            list2=list2->next;
        }
        ListNode* temp = head;
        while(list1!=NULL && list2!=NULL){
            if(list1->val<list2->val){
                temp->next = list1;
                list1=list1->next;
                temp=temp->next;
                continue;
            }
            temp->next = list2;
            list2=list2->next;
            temp=temp->next;
        }
        while(list1!=NULL){
            temp->next = list1;
            list1=list1->next;
            temp=temp->next;
        }
        while(list2!=NULL){
            temp->next = list2;
            list2=list2->next;
            temp=temp->next;
        }
        return head;
    }
};
```