MCS – 253P ADVANCED PROGRAMMING AND PROBLEM SOLVING

LAB 2 PROGRAM (4SUM)

Aswin Sampath
saswin@uci.edu
53844684

Question:

Description | Editorial | Solutions (4K) | Submissions

18. 4Sum

Medium 10.6K 1.3K

Companies

Given an array `nums` of `n` integers, return *an array of all the **unique** quadruplets* `[nums[a], nums[b], nums[c], nums[d]]` such that:

- `0 <= a, b, c, d < n`
- `a`, `b`, `c`, and `d` are **distinct**.
- `nums[a] + nums[b] + nums[c] + nums[d] == target`

You may return the answer in **any order**.

Example 1:

```
Input: nums = [1,0,-1,0,-2,2], target = 0
Output: [[-2,-1,1,2],[-2,0,0,2],[-1,0,0,1]]
```

Example 2:

```
Input: nums = [2,2,2,2,2], target = 8
Output: [[2,2,2,2]]
```

Constraints:

- `1 <= nums.length <= 200`
- `-10^9 <= nums[i] <= 10^9`
- `-10^9 <= target <= 10^9`

Code:

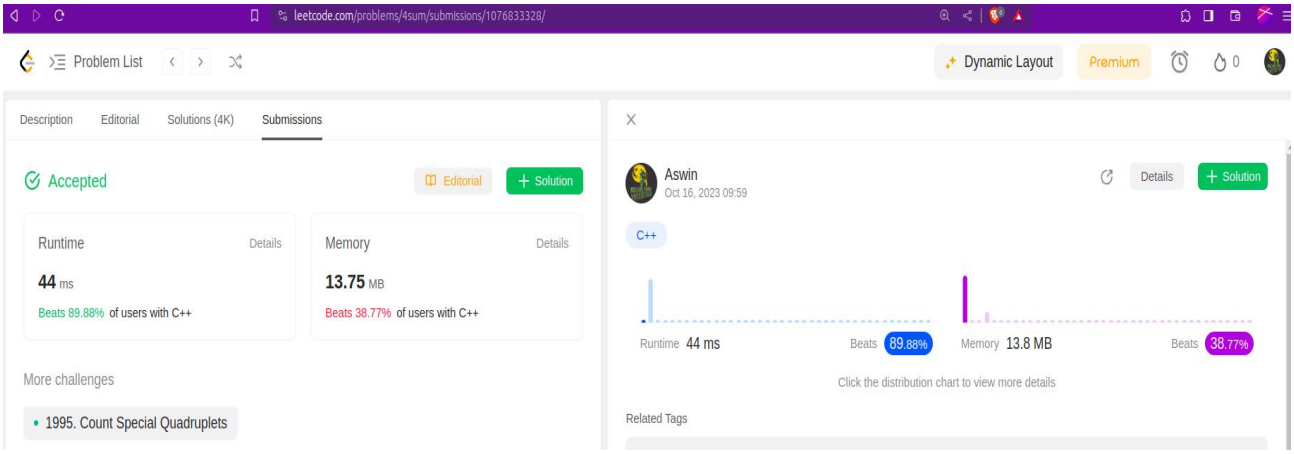
```cpp
class Solution {
public:
    string reorganizeString(string s) {
        string ans;
        int n = s.size();
        priority_queue<pair<int,char>>pq;
        unordered_map<char,int>um;

        for(char c:s){
            um[c]++;
        }

        for(auto p:um){
            pq.push(
                make_pair(
                    p.second,
                    p.first
                )
            );
        }

        while(!pq.empty()){
            // Get the top ele and add it to ans
            pair<int,char> topEle = pq.top();
            pq.pop();
            ans+=topEle.second;
            if(!pq.empty()){
                pair<int,char> secondEle = pq.top();
                pq.pop();
                ans+=secondEle.second;
                if(secondEle.first>1){
                    pq.push(
                        make_pair(
                            --secondEle.first,
                            secondEle.second
                        )
                    );
                }
            }
            if(topEle.first>1){
                pq.push(
                    make_pair(
                        --topEle.first,
                        topEle.second
                    )
                );
            }
        }
        for(int i=1;i<n;i++){
            if(ans[i]==ans[i-1])return "";
        }
        return ans;
    }
};
```

Output:


leetcode.com/problems/4sum/submissions/1076833328/
Problem List
Dynamic Layout
Premium
Description
Editorial
Solutions (4K)
Submissions
Accepted
Editorial
Solution
Runtime
Details
44 ms
Beats 89.88% of users with C++
Memory
Details
13.75 MB
Beats 38.77% of users with C++
More challenges
1995. Count Special Quadruplets
Aswin
Oct 16, 2023 09:59
Details
Solution
C++
Runtime 44 ms
Beats 89.88%
Memory 13.8 MB
Beats 38.77%
Click the distribution chart to view more details
Related Tags