

451. Sort Characters By Frequency

Solution:-

```
class Solution {
public:
    string frequencySort(string s) {

        unordered_map<char,int>freq;
        string str="";

        for(auto & it:s){
            freq[it]++;
        }
        // for(auto it:freq){
        //     cout<<it.first<<"---"><<it.second<<endl;
        //     for(int i=0;i<it.second;i++){
        //         str+=it.first;
        //     }
        // }

        while(!freq.empty()){
            int maxi=INT_MIN;
            char ch;

            for(auto& it:freq){
                maxi=max(it.second,maxi);
            }
            for(auto& it:freq){
```

```

        if(it.second==maxi) ch=it.first;
    }
    for(int i=0;i<maxi;i++){
        str+=ch;
    }
    freq.erase(ch);
}

return str;
}
};

```

The screenshot displays a LeetCode submission page for a problem. The top navigation bar includes logos for Microsoft, Facebook, Amazon, Bloomberg, and Apple. The main interface is divided into several sections:

- Problem List:** Shows the problem status as "Accepted".
- Submissions:** Indicates the submission was accepted by "Rahul kumar" on Mar 11, 2025, at 11:41.
- Runtime:** Shows a runtime of 12 ms, which is 6.61% faster than other submissions.
- Memory:** Shows a memory usage of 11.00 MB, which is 59.93% better than other submissions.
- Code:** Displays the C++ code for the solution, which uses an unordered map to count character frequencies and then constructs the result string by repeating each character according to its frequency.
- Testcase:** Shows the input string "tree" and the output "etree".