

## 451. Sort Characters by Frequency

Given a string  $s$ , sort in decreasing order based on the frequency of characters. Frequency is the number of times it occurs in a string.

Return the sorted string. If multiple answers, return any valid string.

Ex. 1)

Input:  $s = \text{"tree"}$

Output:  $\text{"eert"}$  or  $\text{"eetr"}$

Process:

- 1.) Use a hash map to count occurrence of each unique character.
- 2.) Use an ordered multimap or map in step 1 to flip key and value.
- 3.) Iterate over each key value in multimap, appending  $\text{value} * \text{key}$  to end of result string.
- 4.) Result string contents is answer.

```
std::string frequencySort(const std::string& s) {  
    std::unordered_map<char, int> m;  
    for (const auto c: s) {  
        m[c] += 1;  
    }  
    std::multimap<int, char> m1;  
    for (const auto& [key, value]: m) {  
        m1.insert({value, key});  
    }  
}
```

```
std::string result;  
for (const auto & [key, value]: m1) {  
    result += std::string(key, value);  
}  
return result;
```

```
}
```