Description | Accepted | Editorial | Solutions | Submissions

451. Sort Characters By Frequency

Solved

Medium | Topics | Companies

Given a string `s`, sort it in **decreasing order** based on the **frequency** of the characters. The **frequency** of a character is the number of times it appears in the string.

Return *the sorted string*. If there are multiple answers, return *any of them*.

Example 1:

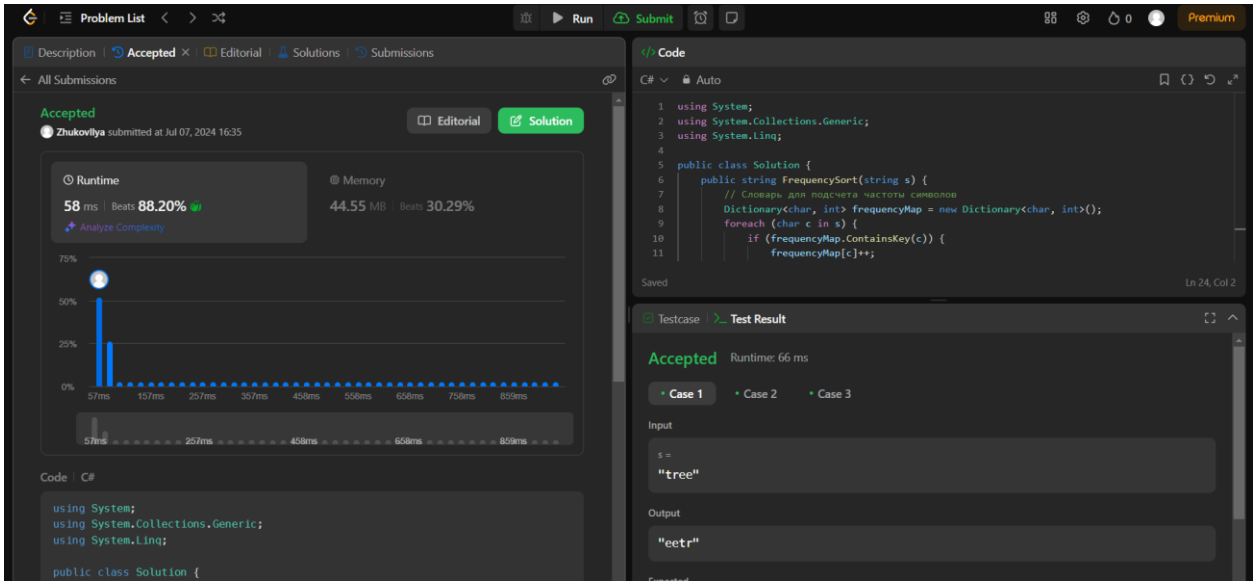
> **Input:** s = "tree"
> **Output:** "eert"
> **Explanation:** 'e' appears twice while 'r' and 't' both appear once.
> So 'e' must appear before both 'r' and 't'. Therefore "eetr" is also a valid answer.

Example 2:

> **Input:** s = "cccaaa"
> **Output:** "aaaccc"
> **Explanation:** Both 'c' and 'a' appear three times, so both "cccaaa" and "aaaccc" are valid answers.
> Note that "cacaca" is incorrect, as the same characters must be together.

Example 3:

> **Input:** s = "Aabb"



Код:

```
using System;
using System.Collections.Generic;
using System.Linq;

public class Solution
```

```
{
    public string FrequencySort(string s)
    {
        // Словарь для подсчета частоты символов
        Dictionary<char, int> frequencyMap = new Dictionary<char, int>();
        foreach (char c in s)
        {
            if (frequencyMap.ContainsKey(c))
            {
                frequencyMap[c]++;
            }
            else
            {
                frequencyMap[c] = 1;
            }
        }

        // Сортировка символов по частоте
        var sortedChars = frequencyMap.OrderByDescending(pair => pair.Value)
                                      .SelectMany(pair => new string(pair.Key,
pair.Value));

        // Объединение отсортированных символов в строку
        return new string(sortedChars.ToArray());
    }
}
```