51. Sort Characters By Frequency

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.

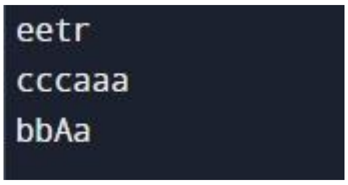
Example 2: Input: s = "cccaaa" Output: "aaaccc" Explanation: Both 'c' and 'a' appear three times, so both "cccaaa" and "aaaccc" are valid answers. Note that "cacaca" is incorrect, as the same characters must be together.

Example 3: Input: s = "Aabb" Output: "bbAa" Explanation: "bbaA" is also a valid answer, but "Aabb" is incorrect. Note that 'A' and 'a' are treated as two different characters.

AIM: Tp sort Characters by Frequency

PROGRAM:

```
def frequencySort(s):
    char_freq = {}
    for char in s:
        char_freq[char] = char_freq.get(char, 0) + 1
    sorted_chars = sorted(char_freq.keys(), key=lambda x: char_freq[x], reverse=True)
    sorted_str = ''.join([char * char_freq[char] for char in sorted_chars])
    return sorted_str
s1 = "tree"
s2 = "cccaaa"
s3 = "Aabb"
print(frequencySort(s1))
print(frequencySort(s2))
print(frequencySort(s3))
```

OUTPUT: 

```
eetr
cccaaa
bbAa
```

TIME COMPLEXITY: O(n+m log m)