

# Problem 1781: Sum of Beauty of All Substrings

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 72.85%

**Paid Only:** No

**Tags:** Hash Table, String, Counting

## Problem Description

The \*\*beauty\*\* of a string is the difference in frequencies between the most frequent and least frequent characters.

\* For example, the beauty of `"abaacc"` is ` $3 - 1 = 2$ `.

Given a string `s`, return \_the sum of\*\*beauty\*\* of all of its substrings.\_

**Example 1:**

**Input:** s = "aabcb" **Output:** 5 **Explanation:** The substrings with non-zero beauty are ["aab", "aabc", "aabcb", "abcb", "bcb"], each with beauty equal to 1.

**Example 2:**

**Input:** s = "aabcbbaa" **Output:** 17

**Constraints:**

\* `1 <= s.length <= 500` \* `s` consists of only lowercase English letters.

## Code Snippets

**C++:**

```
class Solution {  
public:  
    int beautySum(string s) {  
  
    }  
};
```

**Java:**

```
class Solution {  
public int beautySum(String s) {  
  
}  
}
```

**Python3:**

```
class Solution:  
    def beautySum(self, s: str) -> int:
```