# 题目

给你一个由小写英文字母（'a' 到 'z'）组成的字符串 s。你的任务是找出出现频率 最高 的元音（'a'、'e'、'i'、'o'、'u' 中的一个）和出现频率最高的辅音（除元音以外的所有字母），并返回这两个频数之和。

注意：如果有多个元音或辅音具有相同的最高频率，可以任选其中一个。如果字符串中没有元音或没有辅音，则其频率视为 0。

一个字母 x 的 频率 是它在字符串中出现的次数。

示例 1：

输入: s = "successes"

输出: 6

解释:

元音有：'u' 出现 1 次，'e' 出现 2 次。最大元音频率 = 2。

辅音有：'s' 出现 4 次，'c' 出现 2 次。最大辅音频率 = 4。

输出为 2 + 4 = 6。

示例 2：

输入: s = "aeiaeia"

输出: 3

解释:

元音有：'a' 出现 3 次，'e' 出现 2 次，'i' 出现 2 次。最大元音频率 = 3。

s 中没有辅音。因此，最大辅音频率 = 0。

输出为 3 + 0 = 3。

提示:

1 <= s.length <= 100

s 只包含小写英文字母

# 分析

## 方法一：遍历

思路：

首先我们使用哈希表记录字符串中每种字符的出现次数，然后再遍历所有小写字母，分元音和辅音两种去统计出现的最大次数，最后求和就是答案。

代码：

class Solution {

public:

bool is\_vowel(char c) {

return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';

}

int maxFreqSum(string s) {

unordered\_map<char, int> mp;

for (auto ch : s) {

mp[ch]++;

}

int vowel = 0, consonant = 0;

for (char ch = 'a'; ch <= 'z'; ch++) {

if (is\_vowel(ch)) {

vowel = max(vowel, mp[ch]);

} else {

consonant = max(consonant, mp[ch]);

}

}

return vowel + consonant;

}

};

复杂度分析

时间复杂度：O(n)，其中 n 是字符串 s 的长度。

空间复杂度：O(C)，其中 C 是字符集的大小，这里是 26。