# 题目

如果两个字符串 word1 和 word2 中从 'a' 到 'z' 每一个字母出现频率之差都 不超过 3 ，那么我们称这两个字符串 word1 和 word2 几乎相等 。

给你两个长度都为 n 的字符串 word1 和 word2 ，如果 word1 和 word2 几乎相等 ，请你返回 true ，否则返回 false 。

一个字母 x 的出现 频率 指的是它在字符串中出现的次数。

示例 1：

输入：word1 = "aaaa", word2 = "bccb"

输出：false

解释：字符串 "aaaa" 中有 4 个 'a' ，但是 "bccb" 中有 0 个 'a' 。

两者之差为4，大于上限 3 。

示例 2：

输入：word1 = "abcdeef", word2 = "abaaacc"

输出：true

解释：word1 和 word2 中每个字母出现频率之差至多为 3 ：

- 'a'在word1中出现了1次，在word2中出现了4次，差为 3 。

- 'b'在word1中出现了1次，在word2中出现了1次，差为 0 。

- 'c'在word1中出现了1次，在word2中出现了2次，差为1。

- 'd'在word1中出现了1次，在word2中出现了0次，差为1 。

- 'e'在word1中出现了2次，在word2中出现了0次，差为 2 。

- 'f'在word1中出现了1次，在word2中出现了0次，差为 1 。

示例 3：

输入：word1 = "cccddabba", word2 = "babababab"

输出：true

解释：word1 和 word2 中每个字母出现频率之差至多为 3 ：

- 'a'在word1中出现了2次，在word2中出现了4次，差为 2 。

- 'b' word1中出现了2 次，在 word2 中出现了 5 次，差为 3。

- 'c'在word1中出现了3次，在word2中出现了0次，差为3 。

- 'd'在word1中出现了2次，在word2中出现了0次，差为2。

提示：

n == word1.length == word2.length

1 <= n <= 100

word1 和 word2 都只包含小写英文字母。

# 分析

## 方法一：哈希表

思路：

首先需要统计两个字符串中每个字母的出现次数，然后对每个字母的出现次数进行比较，判断它们之间的差值是否小于等于3。下面是实现的一个思路：

1、创建两个数组或哈希表，分别用于存储word1和word2中每个字母的出现次数。

2、遍历word1和word2，统计每个字母的出现次数。

3、对于每个字母，计算其在两个字符串中出现次数的差值，如果差值超过3，则返回false。

4、如果遍历完成后没有发现任何字母的差值超过3，则返回true。

代码：

class Solution {

public:

bool checkAlmostEquivalent(string word1, string word2) {

vector<int> count1(26, 0), count2(26, 0); // 用于存储字母出现次数的数组

// 统计word1中每个字母的出现次数

for (char c : word1) {

count1[c - 'a']++;

}

// 统计word2中每个字母的出现次数

for (char c : word2) {

count2[c - 'a']++;

}

// 检查每个字母的出现次数差值是否超过3

for (int i = 0; i < 26; i++) {

if (abs(count1[i] - count2[i]) > 3) {

return false;

}

}

return true;

}

};

或：

class Solution {

public:

bool checkAlmostEquivalent(string word1, string word2) {

unordered\_map<char, int> freq; // 频数差哈希表

for (auto ch: word1){

++freq[ch];

}

for (auto ch: word2){

--freq[ch];

}

// 判断每个字符频数差是否均小于等于 3

return all\_of(freq.cbegin(), freq.cend(), [](auto&& x) { return abs(x.second) <= 3; });

}

};

或：

class Solution {

public:

bool checkAlmostEquivalent(string word1, string word2) {

unordered\_map<char, int> freq; // 频数差哈希表

for (auto ch: word1){

++freq[ch];

}

for (auto ch: word2){

--freq[ch];

}

// 判断每个字符频数差是否均小于等于 3

for (auto&& x : freq) {

if (abs(x.second) > 3) {

return false;

}

}

return true;

}

};