# 1657. Determine if Two Strings Are Close

Two strings are considered close if you can attain one from the other using the following operations:

Operation 1: Swap any two existing characters.

For example, abcde -> aecdb

Operation 2: Transform every occurrence of one existing character into another existing character, and do the same with the other character.

For example, aacabb -> bbcbaa (all a's turn into b's, and all b's turn into a's)

You can use the operations on either string as many times as necessary.

Given two strings, word1 and word2, return true if word1 and word2 are close, and false otherwise.

## SOLUTION IN C++

class Solution {

public:

bool closeStrings(string word1, string word2) {

if (word1.length() != word2.length())

return false;

unordered\_map<char, int> count1;

unordered\_map<char, int> count2;

string s1; // Unique chars in word1

string s2; // Unique chars in word2

vector<int> freqs1; // Freqs of unique chars in word1

vector<int> freqs2; // Freqs of unique chars in word2

for (const char c : word1)

++count1[c];

for (const char c : word2)

++count2[c];

for (const auto& [c, freq] : count1) {

s1 += c;

freqs1.push\_back(freq);

}

for (const auto& [c, freq] : count2) {

s2 += c;

freqs2.push\_back(freq);

}

ranges::sort(s1);

ranges::sort(s2);

if (s1 != s2)

return false;

ranges::sort(freqs1);

ranges::sort(freqs2);

return freqs1 == freqs2;

}

};