75 Days of Code Day 22 Problem no: 1657. Determine if Two Strings Are Close (leetcode)
Type Hashmap /Sets

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.

Example 1:

Input: word1 = "abc", word2 = "bca"
Output: true
Explanation: You can attain word2 from word1 in 2 operations.
Apply Operation 1: "abc" -> "acb"
Apply Operation 1: "acb" -> "bca"
Example 2:

Input: word1 = "a", word2 = "aa"
Output: false

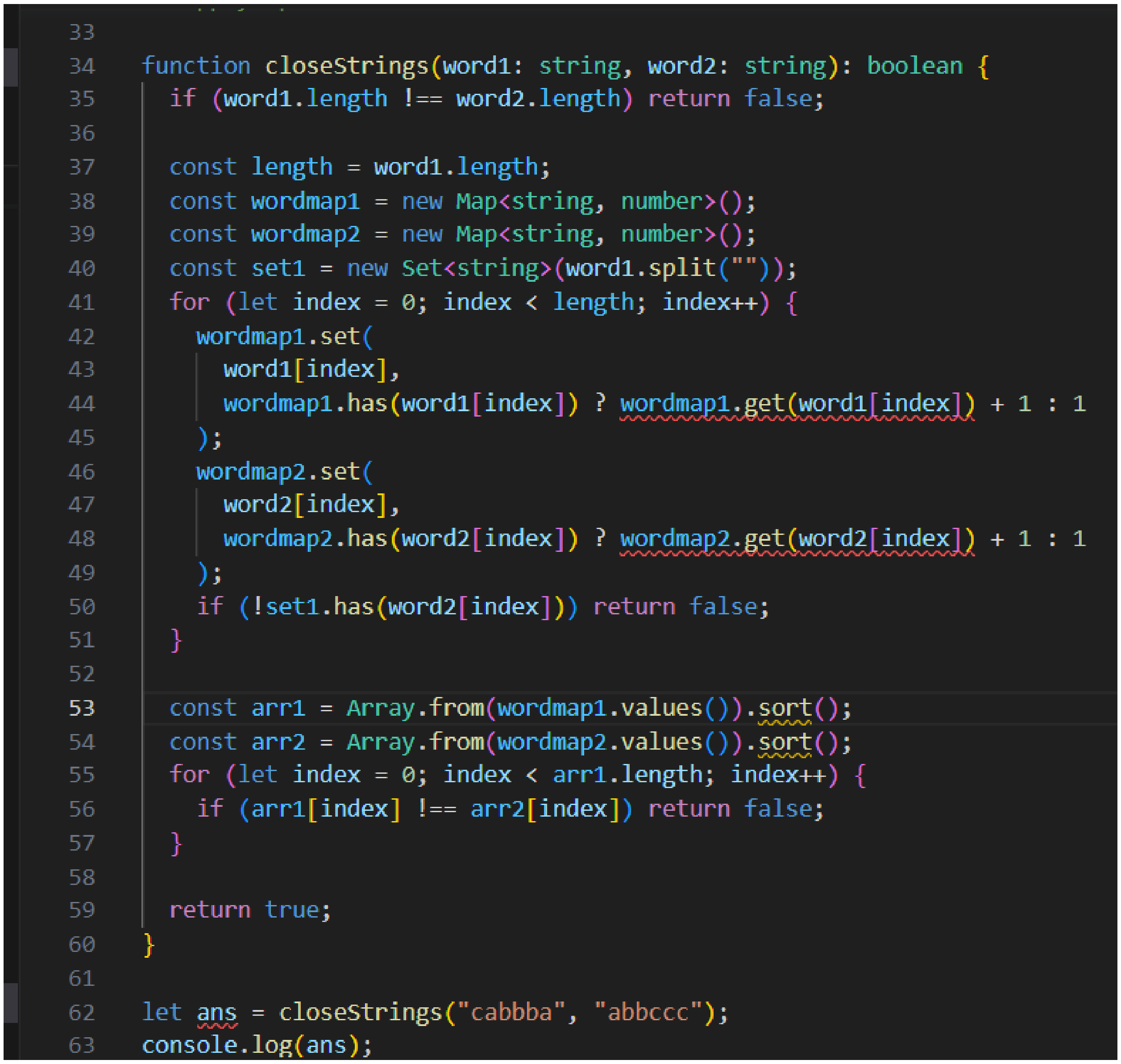
Shubham Agrahari

Explanation: It is impossible to attain word2 from word1, or vice versa, in any number of operations.
Example 3:

Input: word1 = "cabbba", word2 = "abbccc"
Output: true
Explanation: You can attain word2 from word1 in 3 operations.
Apply Operation 1: "cabbba" -> "caabbb"
Apply Operation 2: "caabbb" -> "baaccc"
Apply Operation 2: "baaccc" -> "abbccc"

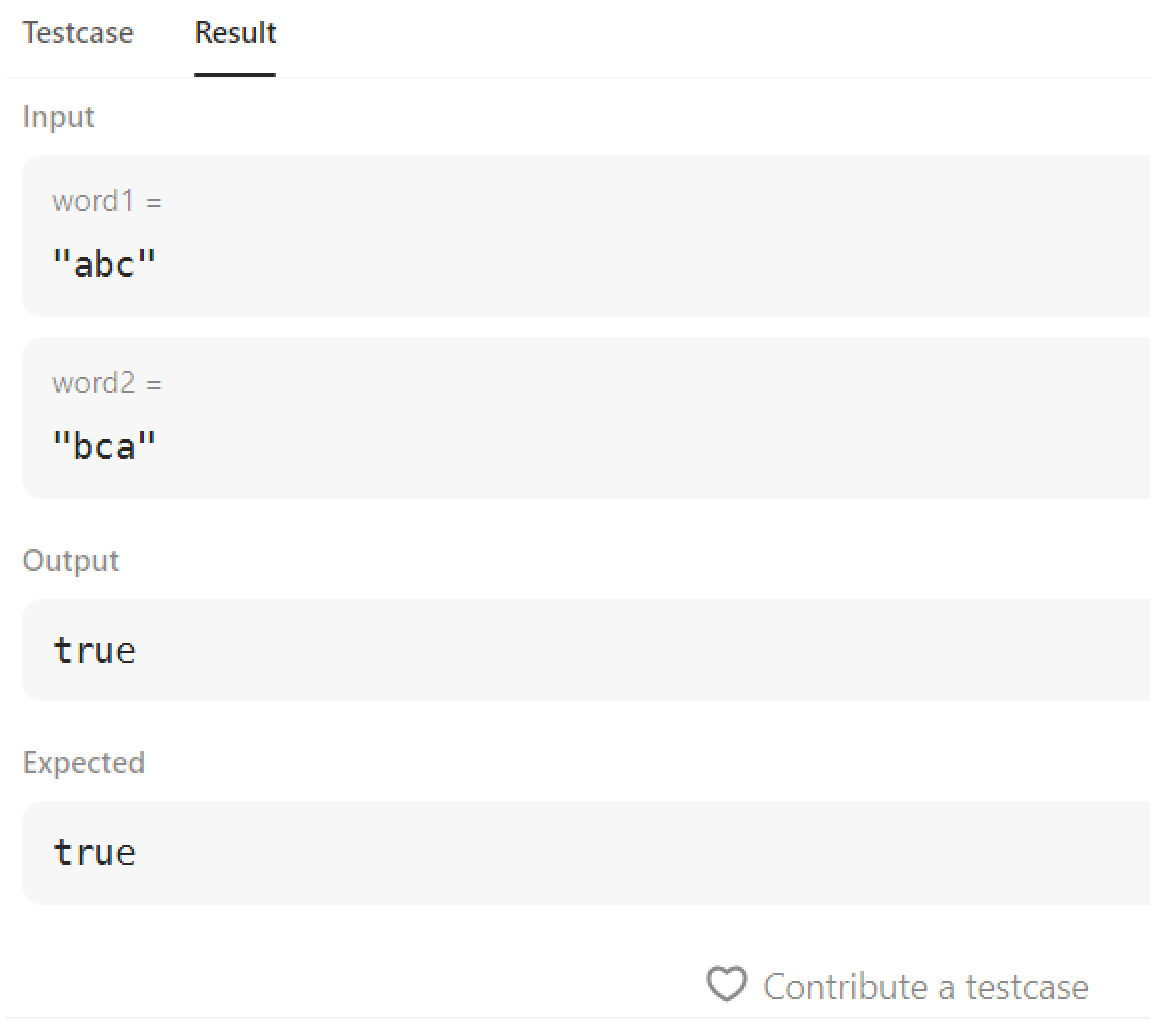
Solution of above problem using in Optimize way using map and set

1. Using logic, we list out all the possible logical ways this can either be true or false.
2. Length of two strings must be equal
3. They should not contain different characters, they must have the same characters, the number of occurrences or the count of each character does not matter.
4. The two arrays arr1 and arr2 of character occurence's count, when sorted, should be equal.



```typescript

function closeStrings(word1: string, word2: string): boolean {
  if (word1.length !== word2.length) return false;

  const length = word1.length;
  const wordmap1 = new Map<string, number>();
  const wordmap2 = new Map<string, number>();
  const set1 = new Set<string>(word1.split(""));
  for (let index = 0; index < length; index++) {
    wordmap1.set(
      word1[index],
      wordmap1.has(word1[index]) ? wordmap1.get(word1[index]) + 1 : 1
    );
    wordmap2.set(
      word2[index],
      wordmap2.has(word2[index]) ? wordmap2.get(word2[index]) + 1 : 1
    );
    if (!set1.has(word2[index])) return false;
  }

  const arr1 = Array.from(wordmap1.values()).sort();
  const arr2 = Array.from(wordmap2.values()).sort();
  for (let index = 0; index < arr1.length; index++) {
    if (arr1[index] !== arr2[index]) return false;
  }

  return true;
}

let ans = closeStrings("cabbba", "abbccc");
console.log(ans);
```

Shubham Agrahari



word1 =
```
"abc"
```
word2 =
```
"bca"
```
Output
```
true
```
Expected
```
true
```

Shubham Agrahari