(/)

7005. Check if Strings Can be Made Equal With Operations II

contest/biweekly-contest-112/problems/check-if-strings-can-be-made-equal-with-operations-ii/submissions/)

ontest/biweekly-contest-112/)

You are given two strings `s1` and `s2`, both of length `n`, consisting of **lowercase** English letters.

You can apply the following operation on **any** of the two strings **any** number of times:

- Choose any two indices `i` and `j` such that `i < j` and the difference `j - i` is **even**, then **swap** the two characters at those indices in the string.

Return `true` *if you can make the strings* `s1` *and* `s2` *equal, and* `false` *otherwise*.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: s1 = "abcdba", s2 = "cabdab"
Output: true
Explanation: We can apply the following operations on s1:
- Choose the indices i = 0, j = 2. The resulting string is s1 = "cbadba".
- Choose the indices i = 2, j = 4. The resulting string is s1 = "cbbdaa".
- Choose the indices i = 1, j = 5. The resulting string is s1 = "cabdab" = s2.
```

Example 2:

```
Input: s1 = "abe", s2 = "bea"
Output: false
Explanation: It is not possible to make the two strings equal.
```

Constraints:

- `n == s1.length == s2.length`
- `1 <= n <= 10^5`
- `s1` and `s2` consist only of lowercase English letters.

JavaScript

```javascript
const checkStrings = (s, t) => {
    let n = s.length, sa = [], sb = [], ta = [], tb = [];
    for (let i = 0; i < n; i++) {
        if (i % 2 == 0) {
            sa.push(s[i]);
            ta.push(t[i]);
        } else {
            sb.push(s[i]);
            tb.push(t[i]);
        }
    }
    sa = sa.sort().join("");
    sb = sb.sort().join("");
    ta = ta.sort().join("");
    tb = tb.sort().join("");
    return sa + sb == ta + tb;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/1038598934/)

More Details > (/submissions/detail/1038598934/)

Share your acceptance!