Problem 1433: Check If a String Can Break Another String

Problem Information

Difficulty: Medium
Acceptance Rate: 70.72%
Paid Only: No
Tags: String, Greedy, Sorting

Problem Description

Given two strings: `s1` and `s2` with the same size, check if some permutation of string `s1` can break some permutation of string `s2` or vice-versa. In other words `s2` can break `s1` or vice-versa.

A string `x` can break string `y` (both of size `n`) if `x[i] >= y[i]` (in alphabetical order) for all `i` between `0` and `n-1`.

Example 1:

Input: s1 = "abc", s2 = "xya" **Output:** true **Explanation:** "ayx" is a permutation of s2="xya" which can break to string "abc" which is a permutation of s1="abc".

Example 2:

Input: s1 = "abe", s2 = "acd" **Output:** false **Explanation:** All permutations for s1="abe" are: "abe", "aeb", "bae", "bea", "eab" and "eba" and all permutation for s2="acd" are: "acd", "adc", "cad", "cda", "dac" and "dca". However, there is not any permutation from s1 which can break some permutation from s2 and vice-versa.

Example 3:

Input: s1 = "leetcodee", s2 = "interview" **Output:** true

Constraints:

* `s1.length == n` * `s2.length == n` * `1 <= n <= 10^5` * All strings consist of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
bool checkIfCanBreak(string s1, string s2) {

}
};
```

Java:

```
class Solution {
public boolean checkIfCanBreak(String s1, String s2) {

}
}
```

Python3:

```
class Solution:
def checkIfCanBreak(self, s1: str, s2: str) -> bool:
```