22. 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.

Code:

```python
def can_break(s1, s2):
    sorted_s1 = sorted(s1)
    sorted_s2 = sorted(s2)

    def can_first_break_second(first, second):
        return all(f >= s for f, s in zip(first, second))

    return can_first_break_second(sorted_s1, sorted_s2) or can_first_break_second(sorted_s2, sorted_s1)


s1 = "abc"
s2 = "xya"
print(can_break(s1, s2))

s1 = "abe"
s2 = "acd"
print(can_break(s1, s2))
```

Output:

```
True
False
```

Time Complexity:

- T(n)= O(n)