Problem 791: Custom Sort String

Problem Information

Difficulty: Medium
Acceptance Rate: 72.25%
Paid Only: No
Tags: Hash Table, String, Sorting

Problem Description

You are given two strings `order` and `s`. All the characters of `order` are **unique** and were sorted in some custom order previously.

Permute the characters of `s` so that they match the order that `order` was sorted. More specifically, if a character `x` occurs before a character `y` in `order`, then `x` should occur before `y` in the permuted string.

Return _any permutation of_`s` _that satisfies this property_.

Example 1:

Input: order = "cba", s = "abcd"

Output: "cbad"

Explanation: `"a"`, `"b"`, `"c"` appear in order, so the order of `"a"`, `"b"`, `"c"` should be `"c"`, `"b"`, and `"a"`.

Since `"d"` does not appear in `order`, it can be at any position in the returned string. `"dcba"`, `"cdba"`, `"cbda"` are also valid outputs.

Example 2:

Input: order = "bcafg", s = "abcd"

Output: "bcad"

Explanation: The characters `"b"`, `"c"`, and `"a"` from `order` dictate the order for the characters in `s`. The character `"d"` in `s` does not appear in `order`, so its position is flexible.

Following the order of appearance in `order`, `"b"`, `"c"`, and `"a"` from `s` should be arranged as `"b"`, `"c"`, `"a"`. `"d"` can be placed at any position since it's not in order. The output `"bcad"` correctly follows this rule. Other arrangements like `"dbca"` or `"bcda"` would also be valid, as long as `"b"`, `"c"`, `"a"` maintain their order.

Constraints:

* `1 <= order.length <= 26` * `1 <= s.length <= 200` * `order` and `s` consist of lowercase English letters. * All the characters of `order` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
string customSortString(string order, string s) {

}
};
```

Java:

```
class Solution {
public String customSortString(String order, String s) {

}
}
```

Python3:

```
class Solution:
def customSortString(self, order: str, s: str) -> str:
```