Description | Solution | Discuss (235) | Submissions

291. Word Pattern II

Medium 622 49 Add to List Share

Given a `pattern` and a string `s`, return `true` *if* `s` ***matches*** *the* `pattern`.

A string `s` **matches** a `pattern` if there is some **bijective mapping** of single characters to strings such that if each character in `pattern` is replaced by the string it maps to, then the resulting string is `s`. A **bijective mapping** means that no two characters map to the same string, and no character maps to two different strings.

Example 1:

```
Input: pattern = "abab", s = "redblueredblue"
Output: true
Explanation: One possible mapping is as follows:
'a' -> "red"
'b' -> "blue"
```

Example 2:

```
Input: pattern = "aaaa", s = "asdasdasdasd"
Output: true
Explanation: One possible mapping is as follows:
'a' -> "asd"
```

Example 3:

```
Input: pattern = "abab", s = "asdasdasdasd"
Output: true
Explanation: One possible mapping is as follows:
'a' -> "a"
'b' -> "sdasd"
Note that 'a' and 'b' cannot both map to "asd" since the mapping is a bijection.
```

Example 4:

```
Input: pattern = "aabb", s = "xyzabcxzyabc"
Output: false
```

Constraints:

- `1 <= pattern.length, s.length <= 20`
- `pattern` and `s` consist of only lower-case English letters.

Accepted 55,171 | Submissions 121,353

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months | 6 months ~ 1 year | 1 year ~ 2 years

Uber | 3 Bolt | 2

Related Topics

Hash Table String Backtracking

Similar Questions

Word Pattern Easy

Java | Autocomplete

```java
class Solution {
    public boolean wordPatternMatch(String pattern, String s) {

    }
}
```

Problems | Pick One | Prev | 99/99 | Next | Console | Contribute | Run Code | Submit