Problem 291: Word Pattern II

Problem Information

Difficulty: Medium
Acceptance Rate: 48.72%
Paid Only: Yes
Tags: Hash Table, String, Backtracking

Problem Description

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 **non-empty** 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 = "aabb", s = "xyzabcxzyabc" **Output:** false

Constraints:

* `1 <= pattern.length, s.length <= 20` * `pattern` and `s` consist of only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
bool wordPatternMatch(string pattern, string s) {

}
};
```

Java:

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

}
}
```

Python3:

```
class Solution:
def wordPatternMatch(self, pattern: str, s: str) -> bool:
```