单词规律

给定一种规律 pattern 和一个字符串 str ，判断 str 是否遵循相同的规律。

这里的 **遵循** 指完全匹配，例如， pattern 里的每个字母和字符串 str 中的每个非空单词之间存在着双向连接的对应规律。

示例 1:

输入: pattern = "abba", str = "dog cat cat dog"

输出: true

示例 2:

输入:pattern = "abba", str = "dog cat cat fish"

输出: false

示例 3:

输入: pattern = "aaaa", str = "dog cat cat dog"

输出: false

示例 4:

输入: pattern = "abba", str = "dog dog dog dog"

输出: false

```
        //单词分割存储
        //建立双向映射关系进行判断
class Solution {
public:
    bool wordPattern(string pattern, string s) {
        vector<string> v1;
        stringstream s1(s);
        string s2;
        map<char, string> map1;
        map<string, char> map2;

        while (getline(s1, s2, ' '))
        {
            v1.push_back(s2);
        }
        if (pattern.size() != v1.size())
        {
```

```
            return false;
        }

        for (int i = 0; i < v1.size(); i++)
        {
            if (map1.count(pattern[i]) > 0 || map2.count(v1[i]) > 0)
            {
                if (v1[i] != map1[pattern[i]] || map2[v1[i]] != pattern[i])
                {
                    return false;
                }
                continue;
            }
            map1.insert(make_pair(pattern[i], v1[i]));
            map2.insert(make_pair(v1[i], pattern[i]));
        }

        return true;


    }
};
```