

## 290 Word pattern

2018年4月8日 21:21

### Question:

Given a pattern and a string str, find if str follows the same pattern.

Here **follow** means a full match, such that there is a bijection between a letter in pattern and a **non-empty** word in str.

### Examples:

1. pattern = "abba", str = "dog cat cat dog" should return true.
2. pattern = "abba", str = "dog cat cat fish" should return false.
3. pattern = "aaaa", str = "dog cat cat dog" should return false.
4. pattern = "abba", str = "dog dog dog dog" should return false.

### Notes:

You may assume pattern contains only lowercase letters, and str contains lowercase letters separated by a single space.

来自 <<https://leetcode.com/problems/word-pattern/description/>>

给定一种 pattern(模式) 和一个字符串 str，判断 str 是否遵循这种模式。

这里的 **遵循** 指完全匹配，例如在pattern里的每个字母和字符串 str 中的每个非空单词存在双向单映射关系。

### 例如:

1. pattern = "abba", str = "dog cat cat dog", 返回true
2. pattern = "abba", str = "dog cat cat fish", 返回false.
3. pattern = "aaaa", str = "dog cat cat dog", 返回false.
4. pattern = "abba", str = "dog dog dog dog", 返回false.

### 说明:

你可以假设 pattern 只包含小写字母， str 包含了由单个空格分开的小写单词。

### Solution for Python3:

```
1 class Solution1(object):
2     def wordPattern(self, pattern, str):
3         """
4         :type pattern: str
5         :type str: str
6         :rtype: bool
7         """
8         s = pattern
9         t = str.split()
10        return list(map(s.find, s)) == list(map(t.index, t))
11
12 class Solution2(object):
13     def wordPattern(self, pattern, str):
14         """
15         :type pattern: str
16         :type str: str
17         :rtype: bool
18         """
19         f = lambda s: map({}.setdefault, s, range(len(s)))
20        return list(f(pattern)) == list(f(str.split()))
21
22 class Solution3(object):
23     def wordPattern(self, pattern, str):
24         """
25         :type pattern: str
26         :type str: str
27         :rtype: bool
```

```

28         """
29         s = pattern
30         t = str.split()
31         return len(s) == len(t) and len(set(zip(s,t))) == len(set(s)) == len(set(t))

```

## Solution for C++:

```

1  class Solution {
2  public:
3      bool wordPattern(string pattern, string str) {
4          int p2i[26] = {};
5          unordered_map<string, int> w2i;
6          istringstream in(str);
7          int i = 0, n = pattern.size();
8          for (string word; in >> word; ++i) {
9              if (i == n || p2i[pattern[i] - 'a'] != w2i[word])
10                 return false;
11                 p2i[pattern[i] - 'a'] = w2i[word] = i + 1;
12             }
13             return i == n;
14         }
15     };

```

## Appendix:

### Python index()

- 1) index() 函数用于从列表找出某个值第一个匹配项的索引位置。
  - a. [1,23].index(3) -> 2
- 2) Python index() 方法检测字符串中是否包含子字符串 str，如果指定 beg（开始）和 end（结束）范围，则检查是否包含在指定范围内，该方法与 python find()方法一样，只不过如果str不在 string中会报一个异常。
  - a. str.index(str, beg=0, end=len(string))

### Python 字典setdefault()

- 1) Python 字典 setdefault() 方法和 [get\(\)](#) 方法类似,返回指定键的值，如果键不在字典中，将会添加键并将值设置为一个指定值，默认为None。
- 2) get() 和 setdefault() 区别： setdefault() 返回的键如果不在字典中，会添加键（更新字典），而 get() 不会添加键。