题目：

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.

1.时间：O（N）; 空间：O（N）

```
class Solution {
public:
    bool wordPattern(string pattern, string str) {
        if (pattern.empty()) return false;
        if (str.empty()) return true;
```

```
        /* 判断 pattern 大小和 str 里的子字符串数量是否相等 */
        int spaceCount = std::count_if(str.begin(), str.end(), [](const char ch){
            return ch == ' ';
        });
        if (spaceCount + 1 != pattern.size()) return false;

        std::vector<std::string> strs;
        /*for (auto it = str.begin(); it != str.end();){
            auto tmp = std::find(it, str.end(), ' ');
            strs.push_back(str.substr(std::distance(str.begin(),  it),  std::distance(it,
tmp)));
            if (tmp == str.end()) break;
            it = tmp + 1;
        }*/
        int last_spce = 0;
        for (int i = 0; i < str.size(); ++i){
            if (str[i] == ' '){
                strs.push_back(str.substr(last_spce, i - last_spce));
                last_spce = i + 1;
            }
        }
```

```
        strs.push_back(str.substr(last_spce, str.size() - last_spce));


        std::unordered_map<std::string, char> sTable; /* 数据例子：["abba"，"cat dog dog fish"] */
        std::unordered_map<char, std::string> pTable; /*  数据例子：["aaaa", "cat dog dog cat"] */


        for (int i = 0; i < strs.size(); ++i){
            if (sTable.find(strs[i]) != sTable.end() && pTable.find(pattern[i]) != pTable.end()){
                if (sTable[strs[i]] != pattern[i] || pTable[pattern[i]] != strs[i]) return false;    /* 数据例子：["aba"，"cat dog dog"] */
            } else{
                if (sTable.find(strs[i]) != sTable.end() || pTable.find(pattern[i]) != pTable.end()) return false;
                sTable[strs[i]] = pattern[i];
                pTable[pattern[i]] = strs[i];
            }
        }
        return true;
    }
};
```