771 Jewels and Stones

2018年5月4日 17:16

You're given strings J representing the types of stones that are jewels, and S representing the stones you have. Each character in S is a type of stone you have. You want to know how many of the stones you have are also jewels.
The letters in J are guaranteed distinct, and all characters in J and S are letters. Letters are case sensitive, so "a" is considered a different type of stone from "A".
Example 1:
Input: J = "aA", S = "aAAbbbb"
Output: 3
Example 2:
Input: J = "z", S = "ZZ"
Output: 0
Note:

- S and J will consist of letters and have length at most 50.
- The characters in J are distinct.

来自 <https://leetcode.com/problems/jewels-and-stones/description/>
给定字符串J 代表石头中宝石的类型，和字符串 S代表你拥有的石头。 S 中每个字符代表了一种你拥有的石头的类型，你想知道你拥有的石头中有多少是宝石。
J 中的字母不重复，J 和 S中的所有字符都是字母。字母区分大小写，因此"a"和"A"是不同类型的石头。
示例 1:
输入: J = "aA", S = "aAAbbbb"
输出: 3
示例 2:
输入: J = "z", S = "ZZ"
输出: 0
注意:

- S 和 J 最多含有50个字母。
- J 中的字符不重复。

Solution for Python3：

```python
class Solution1:
    def numJewelsInStones(self, J, S):
        """
        :type J: str
        :type S: str
        :rtype: int
        """
        ans = 0
        for s in S:
            if s in J:
                ans += 1
        return ans

class Solution2:
    def numJewelsInStones(self, J, S):
        """
        :type J: str
        :type S: str
        :rtype: int
        """
        return sum(map(J.count, S))
        # return sum(map(S.count, J))
        # return sum(s in J for s in S)
```

Solution for C++:

```cpp
class Solution {
public:
    int numJewelsInStones(string J, string S) {
        int ans = 0;
        for (char s : S) {
            if (J.find(s) != string::npos)
                ans++;
        }
```

```
10              return ans;
11          }
    };
```