771. Jewels and Stones

Easy 1123 203 Favorite Share

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.

Accepted 151,588 | Submissions 184,413

Seen this question in a real interview before? Yes No

Contributor 

```java
package Algorithm;

import java.util.HashMap;

/*
 * 题意：给一个代表不同宝石的字符串，和一个代表石头的字符串，从石头字符串里找出属于宝石的石头，并返回个数
 * 注意，区分大小写
 *
 * 解法：将石头中的每个字符数量进行统计，如果是宝石，则count加上此个数量，如果不好使，则不管
 */
public class L771 {

    public int numJewelsInStones(String J, String S) {

        HashMap<Character, Integer> hashMap = new HashMap<Character, Integer>();

        for(int i = 0; i < S.length(); i ++) {
            if(hashMap.containsKey(S.charAt(i))) {
                hashMap.put(S.charAt(i), hashMap.get(S.charAt(i)) + 1);
            }else {
                hashMap.put(S.charAt(i), 1);
            }
        }
        int count = 0;

        for(int i = 0; i < J.length(); i++) {
            if(hashMap.containsKey(J.charAt(i))) {
                count += hashMap.get(J.charAt(i));
            }
        }

        return count;
    }

}
```