[每日 LeetCode] 20. Valid Parentheses

作者：Hanseltu
原文链接：https://ld246.com/article/1557324943753
来源网站：链滴

原文链接 [每日LeetCode] 20. Valid Parentheses

Description:

Given a string containing just the characters '(' , ')' , '{' , '}' , '['and']' , determine if the input stri g is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

Example 1:

Input: "()"
Output: true

Example 2:

Input: "()[]{}"
Output: true

Example 3:

Input: "(]"
Output: false

Example 4:

Input: "([)]"
Output: false

Example 5:

Input: "{[]}"
Output: true

思路：本题要求判断输入的括号字符串是否是合法的括号，这是一道经典的栈的应用题，讲到栈的地应该都会讲到。正常扫描字符串，遇到左伴侣 '[' '{' '('就压栈，遇到右伴侣 ']' '}' ')'就判断栈顶是否与相匹配，如果匹配则弹出，否则直接返回false。最后判断栈是否为空。比较诡异的是代码提交后运行间是0ms！！！，头一次遇到这种情况。

C++代码

```
class Solution {
public:
    bool isValid(string s) {
        stack<char> st;
```

```
        if (s.size() == 0)
            return true;
        for(int i=0; i<s.size(); i++){
            if (s[i] == '(' || s[i] == '{' || s[i] == '[')
                st.push(s[i]);
            else {
                if(st.empty() || (s[i] == ')' && st.top() != '(')
                   || (s[i] == '}' && st.top() != '{')
                   || (s[i] == ']' && st.top() != '[') )
                    return false;
                st.pop();
            }
        }
        return st.empty();
    }
};
```

运行时间：0ms

运行内存：8.5M