20. 有效的括号

> 亚马逊、JPMorgan 在半年内面试常考

地址

有效的括号

题目

20. 有效的括号

难度 简单 2505 收藏 分享 切换为英文 接收动态 反馈

给定一个只包括 `'('`，`')'`，`'{'`，`'}'`，`'['`，`']'` 的字符串 `s` ，判断字符串是否有效。

有效字符串需满足：

1. 左括号必须用相同类型的右括号闭合。
2. 左括号必须以正确的顺序闭合。

示例 1：

```
输入：s = "()"
输出：true
```

示例 2：

```
输入：s = "()[]{}"
输出：true
```

示例 3：

```
输入：s = "(]"
输出：false
```

示例 4：

```
输入：s = "([)]"
输出：false
```

示例 5：

```
输入：s = "{[]}"
输出：true
```

提示：

- `1 <= s.length <= 10^4`
- `s` 仅由括号 `'()[]{}'` 组成

20. Valid Parentheses

难度 简单 2505 收藏 分享 切换为中文 接收动态 反馈

Given a string `s` containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string 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.

Example 1:

```
Input: s = "()"
Output: true
```

Example 2:

```
Input: s = "()[]{}"
Output: true
```

Example 3:

```
Input: s = "(]"
Output: false
```

Example 4:

```
Input: s = "([)]"
Output: false
```

Example 5:

```
Input: s = "{[]}"
Output: true
```

Constraints:

- `1 <= s.length <= 10^4`
- `s` consists of parentheses only `'()[]{}'`.

思路 1 ： 栈

1. ({[入栈；
2.) }] 与栈顶比较，对应则弹出栈顶.

代码：

```
// Java
// Time : 2021 - 07 - 19

public boolean isValid(String s) {
```

```
        // initialize a stack
        Stack<Character> stack = new Stack<>();

        // 将 string -> char 迭代循环 s
        for (Character c : s.toCharArray()) {

            if (c == '(' || c == '{' || c == '[') {                 // ({[, 入栈
                stack.push(c);
            } else if(c==')'&&!stack.empty()&&stack.peek()=='('){   // ),栈非空,
栈顶元素为 (, 出栈
                stack.pop();
            } else if(c=='}'&&!stack.empty()&&stack.peek()=='{'){  // },栈非空,
栈顶元素为 {, 出栈
                stack.pop();
            } else if(c==']'&&!stack.empty()&&stack.peek()=='['){  // ],栈非空,
栈顶元素为 [, 出栈
                stack.pop();
            } else{
                // c 是 )}],栈空，则无效
                // c 是 )}],栈非空，但不匹配，则无效
                return false;
            }
        }

        // 栈空，全部匹配，括号有效
        // 栈非空，有多余括号，括号无效
        return stack.empty();
    }
```

时间复杂度 ：

- 时间复杂度：需要遍历整个字符串 s, 所以为 **O(n)**;
- 空间复杂度：需要额外堆栈进行空间分配，长度为 n , 所以为 **O (n)**.

思路 2 ： 暴力解法

#Leetcode/Stack#