

0020. 有效的括号

👤 ITCharge ⌚ 大约 2 分钟

- 标签：栈、字符串
- 难度：简单

题目链接

- [0020. 有效的括号 - 力扣](#)

题目大意

描述： 给定一个只包括 '(' , ')' , '{' , '}' , '[' , ']' 的字符串 *s* 。

要求： 判断字符串 *s* 是否有效（即括号是否匹配）。

说明：

- 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：

- 示例 1:

输入: *s* = "()"

输出: **True**

py

- 示例 2:

输入: *s* = "()[]{}"

输出: **True**

py

解题思路

思路 1：栈

括号匹配是「栈」的经典应用。我们可以用栈来解决这道题。具体做法如下：

1. 先判断一下字符串的长度是否为偶数。因为括号是成对出现的，所以字符串的长度应为偶数，可以直接判断长度为奇数的字符串不匹配。如果字符串长度为奇数，则说明字符串 `s` 中的括号不匹配，直接返回 `False`。
2. 使用栈 `stack` 来保存未匹配的左括号。然后依次遍历字符串 `s` 中的每一个字符。
 1. 如果遍历到左括号时，将其入栈。
 2. 如果遍历到右括号时，先看栈顶元素是否是当前右括号相同类型的左括号。
 1. 如果是与当前右括号相同类型的左括号，则令其出栈，继续向前遍历。
 2. 如果不是与当前右括号相同类型的左括号，则说明字符串 `s` 中的括号不匹配，直接返回 `False`。
3. 遍历完，还要再判断一下栈是否为空。
 1. 如果栈为空，则说明字符串 `s` 中的括号匹配，返回 `True`。
 2. 如果栈不为空，则说明字符串 `s` 中的括号不匹配，返回 `False`。

思路 1：代码

```
class Solution:
    def isValid(self, s: str) -> bool:
        if len(s) % 2 == 1:
            return False
        stack = list()
        for ch in s:
            if ch == '(' or ch == '[' or ch == '{':
                stack.append(ch)
            elif ch == ')':
                if len(stack) != 0 and stack[-1] == '(':
                    stack.pop()
                else:
                    return False
            elif ch == ']':
                if len(stack) != 0 and stack[-1] == '[':
                    stack.pop()
```

py

```
        else:
            return False
    elif ch == '}':
        if len(stack) != 0 and stack[-1] == '{':
            stack.pop()
        else:
            return False
    if len(stack) == 0:
        return True
    else:
        return False
```

思路 1：复杂度分析

- 时间复杂度： $O(n)$ 。
- 空间复杂度： $O(n)$ 。