

680. 验证回文字符串 II

难度 简单  264     

给定一个非空字符串 `s`，最多删除一个字符。判断是否能成为回文字符串。

示例 1:

输入: "aba"
输出: True

示例 2:

输入: "abca"
输出: True
解释: 你可以删除c字符。

注意:

1. 字符串只包含从 a-z 的小写字母。字符串的最大长度是50000。

思路:

1 回文子串: 左右对称

利用 `s == s[::-1]` 来判断是否是回文

2 判断是否是回文: 让左右指针往中间走,

a, 左右指针指向的元素相等, 继续往中间走, `left += 1, right -= 1`

b, 左右指针遇到的元素不想等, 进行处理。因为现在`[0, left]`和`[right, len(s)-1]`都已经验证过了, 就不验证了
那么现在只需要验证`s[left+1, right+1]`, 跳过`left`, 或者`[left, right]`, 跳过`right`, 是否是回文子串

```
1 def validPalindrome(self, s: str) -> bool:
2     isPalindrome = lambda s : s == s[::-1]
3
4     left = 0
5     right = len(s) - 1
6     while left < right:
7         if s[left] == s[right]:
8             left += 1
9             right -= 1
10        else:
11            return isPalindrome(s[left + 1:right+1]) or isPalindrome(s[left:right])
12
13    return True
```

时间复杂度 $O(n)$

空间复杂度 $O(n)$