

5. Longest Palindromic Substring

[Question](#)[My Submissions](#)

Total Accepted: **85140** Total Submissions: **390340** Difficulty: **Medium**

Given a string S , find the longest palindromic substring in S . You may assume that the maximum length of S is 1000, and there exists one unique longest palindromic substring.

思路：

对于整个字符串，挨个字符进行分析，以某字符为中心向两边查找，找到以该字符为中心的最长Palindrome保存起来，最后选择保存起来的最大的那个。

注意，有两种情况：基数回文字符串（如 **abcfcb**a）和偶数回文字符串（如 **abccba**），对于偶数最长回文字符串，不能以某字符中心分析，要以相邻两个字符一起分析。

例1-基数最长回文字符串：**abcfcb**de 最长字符串 是 **bcfcb**

从左到右开始分析：

index= 0: 第一个字符**a**, 以它为中心的最长字符串长度为1（也就是它自己）max =1

index= 1: 第二个字符**b**, 以它为中心的最长字符串长度为1（也就是它自己）max =1

index= 2: 第三个字符**c**, 以它为中心的最长字符串长度为1（也就是它自己）max =1

index= 3: 第四个字符**f**, 以它为中心的最长字符串长度为 5（也就是它自己）max =5

....

最后 最长回文字符串长度为5，从 index - (5-1)//2 开始，到 index + 5//2 结束

例2-偶数最长回文字符串：**abccb**def

从左向右分析：

index, index+1 = 0, 1: **a**,**b** 最长回文字符串为长度为0（不是回文字符串）

index, index+1 = 1, 2: **b**, **c**, 最长回文字符串长度为0.

index, index+1 = 2, 3: **c**, **c**, 最长回文字符串长度为 4.

...

最后，最长回文字符串长度为4，从 index - (4-1)//2 开始，到index + 4//2 结束。

代码见下页：

PS: 比较值得注意的是实现的时候很容易出现index错误，一定要注意分析获得index之后如何精确找到起始位置和结束位置。

```
13 class Solution(object):
14     def longestPalindrome(self, s):
15
16         n = len(s)
17         start = end = 0
18
19         for i in range(n):
20             len1 = self.helper(s, i, i)
21             len2 = self.helper(s, i, i+1)
22             maxLen = max(len1, len2)
23
24             if maxLen > end - start:
25                 start = i - (maxLen-1) / 2
26                 end = i + maxLen / 2
27
28         return s[start : end+1]
29
30
31     def helper(self, s, l, r):
32         while l >= 0 and r < len(s) and s[l] == s[r]:
33             l -= 1
34             r += 1
35
36         return r - l - 1
37
```