



链滴

# Longest Palindromic Substring

作者: [yudake](#)

原文链接: <https://ld246.com/article/1519264480016>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 题目描述

Given a string **s**, find the longest palindromic substring in **s**. You may assume that the maximum length of **s** is 1000.

### Example:

Input: "babad"

Output: "bab"

Note: "aba" is also a valid answer.

### Example:

Input: "cbbd"

Output: "bb"

找到最长的回文子序列。

## 解题思路

- 暴力循环;
- 动态规划, 用一个二维数组存储相应位置(i,j)是否相同:
  - 时间复杂度:  $O(n^2)$ ;
  - 空间复杂度:  $O(n^2)$ 。
- 从中心向两边扩展:
  - 时间复杂度:  $O(n^2)$ ;
  - 空间复杂度:  $O(1)$ 。

## 代码

### 代码3

```
class Solution {
    public String longestPalindrome(String s) {
        if (s == null || s.length() == 0)
            return null;
        int lIndex = 0;
        int rIndex = 0;
        for (int i = 0; i < s.length(); i++) {
            int left = i-1;
            int right = i+1;
            while (left >= 0 && right < s.length() && s.charAt(left) == s.charAt(right)) {
                left--;
                right++;
            }
        }
    }
}
```

```
    }
    if (right-left-1 > rIndex-lIndex+1) {
        rIndex = right - 1;
        lIndex = left + 1;
    }
    left = i;
    right = i+1;
    while (left >= 0 && right < s.length() && s.charAt(left) == s.charAt(right)) {
        left--;
        right++;
    }
    if (right-left-1 > rIndex-lIndex+1) {
        rIndex = right - 1;
        lIndex = left + 1;
    }
}
return s.substring(lIndex, rIndex+1);
}
```