

BAT算法面试题(六)- 最大回文子串

文章出自 Hello Code 开发者学习平台 CC老师

获取更新文章/视频 关注公众号:

HelloCode开发者学习平台



小编:今天是我们一起坚持的第六天了!继续加油!

回顾一下,昨天我们学习的是回文动态规划处理方法,但是?难道没有更优秀的解决方案吗?肯定有呀!!!

一.算法题

- 题目

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

- Example

Example1:

Input: "babad"

Output: "bab"

Note: "aba" is also a valid answer.

Example2:

Input: "cbbd"

Output: "bb"

二.算法题解读

- **题目大意:**给定一个字符串S,找出S串中最长的回文子串.你可以假设s的最大长度为1000.

Example1:

输入: "babad"

输出: "bab"

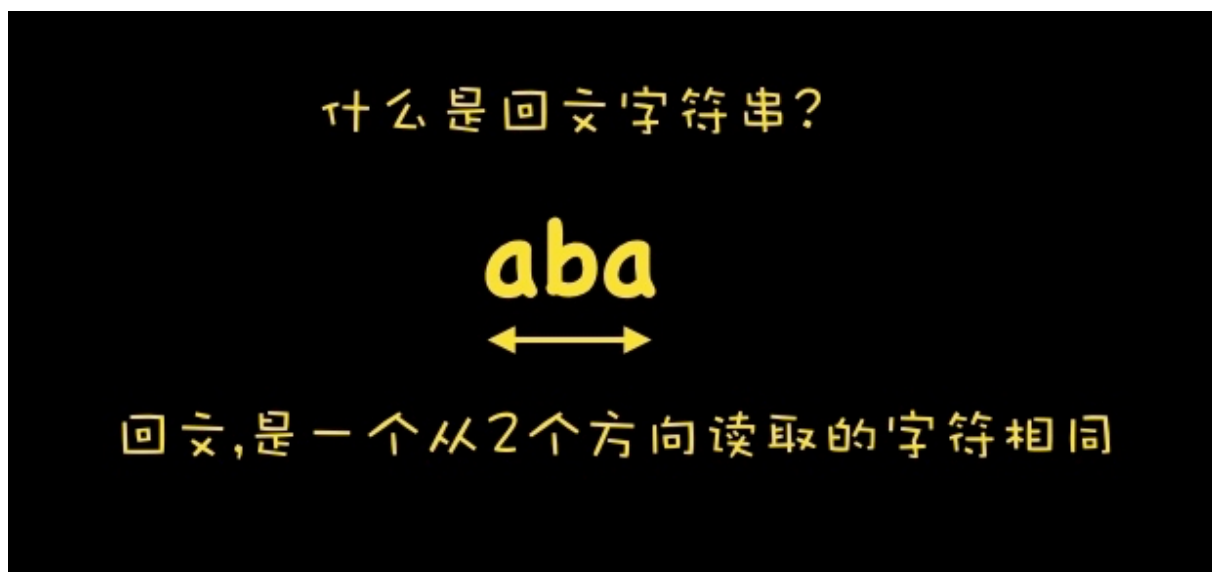
注意: "aba" 是一个有效答案.

Example2:

输入: "cbbd"

输出: "bb"

三.回文字符串



我们上一篇文章分享的不管从时间复杂度还是空间复杂度,都是颇为浪费的? 难道没有更优解决方案?肯定是有的!

四.代码

```
string expandAroundCenter(string s, int c1, int c2) {  
    int l = c1, r = c2;
```

```

    int n = s.length();
    while (l >= 0 && r <= n-1 && s[l] == s[r]) {
        l--;
        r++;
    }
    return s.substr(l+1, r-l-1);
}

string longestPalindromeSimple(string s) {
    int n = s.length();
    if (n == 0) return "";
    //单个字符也是回文字符
    string longest = s.substr(0, 1);
    for (int i = 0; i < n-1; i++) {
        string p1 = expandAroundCenter(s, i, i);
        if (p1.length() > longest.length())
            longest = p1;

        string p2 = expandAroundCenter(s, i, i+1);
        if (p2.length() > longest.length())
            longest = p2;
    }
    return longest;
}

```

五.复杂度

时间复杂度: $O(N*N)$

空间复杂度: $O(1)$

六.学习建议

大家可以画10分钟左右,将代码的模拟执行一遍.即可明白其过程.明天我们会更新一种另外的解决方案哦.