BAT算法面试题(五)- 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



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

什么是回文字符串?

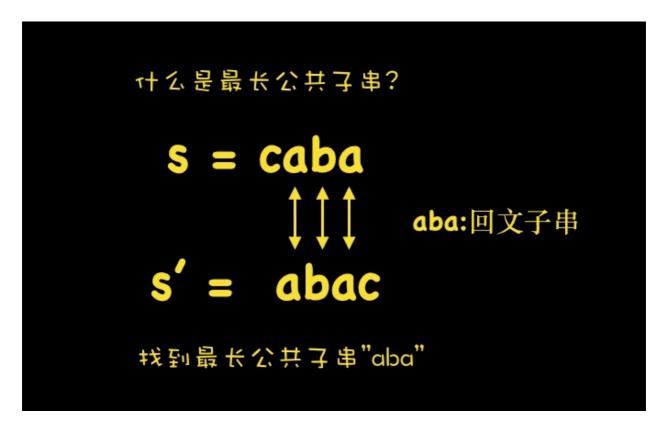


回文,是一个从2个方向读取的字符相同

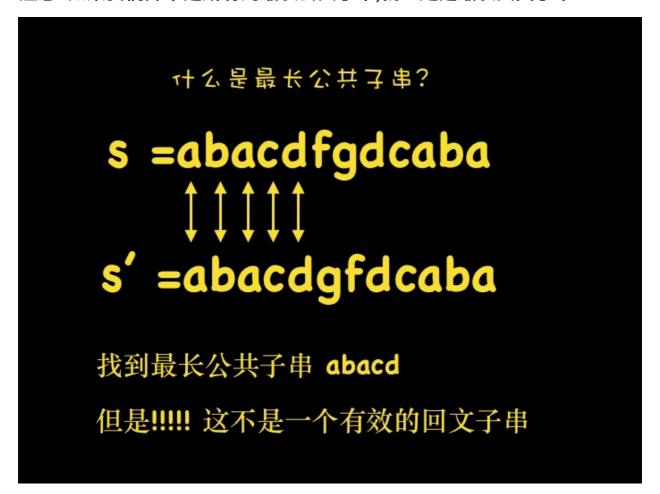
四.找到字符串的最长公共子串

一般开发者,能想到的最快速的方法,就是找到"最长公共子串".

"反转S并成为S',找到S和S'之间的最长公共子串.它也必须是最长的回文子串"



注意: 如果我们并不是所有的最长公共子串,就一定是最长回文子串.



所以,如果只是单纯的查找最长公共子串方法,是不可行的.但是,如果去修改这个问题?

思路:

在我们找到一个最长的公共子串候选者时,我们检查子串的索引是否与反向子串的原始索引相同.如果是,那么尝试更新到目前为止发现的最长的回文. 如果没有,我们就跳过这个,寻找下个候选回文子串.

五.动态编程解决方案

- 如果子串 Si...Sj 是回文,则定义 P[i,j] 为真,否则为假
- 所以, P[i,j] <-- (p[i+1,j-1] 和 Si = Sj);

六.复杂度

时间复杂度:O(N*N)

空间复杂度:O(N*N)

七.代码

C Code

```
string longestPalindromeDP(string s) {
   int n = s.length();
   int longestBegin = 0;
   int maxLen = 1;
   bool table[1000][1000] = {false};
   for (int i = 0; i < n; i++) {
     table[i][i] = true;
   }
   for (int i = 0; i < n-1; i++) {
     if (s[i] == s[i+1]) {
      table[i][i+1] = true;
      longestBegin = i;
      maxLen = 2;
   }
}
for (int len = 3; len <= n; len++) {
   for (int i = 0; i < n-len+1; i++) {</pre>
```

```
int j = i+len-1;
    if (s[i] == s[j] && table[i+1][j-1]) {
        table[i][j] = true;
        longestBegin = i;
        maxLen = len;
    }
    }
}
return s.substr(longestBegin, maxLen);
}
```

八.学习建议

尝试画图->阅读代码

- 算法并不是1+1=2.很多时候需要大家拿好纸笔思考才能感受到它的魅力 之处!
- 此次附上小编学习的时候草稿! 大家也一起吧....

