最长回文子串

给你一个字符串 s，找到 s 中最长的回文子串。

示例 1：

输入：s = "babad"

输出："bab"

解释："aba" 同样是符合题意的答案。

示例 2：

输入：s = "cbbd"

输出："bb"

示例 3：

输入：s = "a"

输出："a"

示例 4：

输入：s = "ac"

输出："a"

字符	b	a	b	a	b
下标	0	1	2	3	4

子串左边界 \ 子串右边界	0	1	2	3	4
0	TRUE	FALSE	TRUE	FALSE	TRUE
1		TRUE	FALSE	TRUE	FALSE
2			TRUE	FALSE	TRUE
3				TRUE	FALSE
4					TRUE

```
class Solution {
public:
    string longestPalindrome(string s) {
```

```
        //动态规划 dp[i][j]表示为子串 s[i,...j]是回文串，包含边界 i 和 j
        int len=s.size();
        if(len<2)
            return s;
        vector<vector<int>> dp(len,vector<int>(len));
        int maxlen=1;//
        int begin=0;
        for(int i=0;i<len;i++)
        {
            dp[i][i]=1;//对角线上  自身单独一个字符肯定为回文
        }
        for(int j=1;j<len;j++)
        {
            for(int i=0;i<j;i++)
            {
                if(s[i]!=s[j])//首先判断头尾是否一致  头尾不一致则无法进行状态转移
                    dp[i][j]=0;
                else
                {
                    if(j-i<3)//保证不越界  头尾相同且回文子串字符个数小于 3  则一定为回文
                        dp[i][j]=1;
                    else
                        dp[i][j]=dp[i+1][j-1];//满足头尾相同，且子串个数 3 个以上采用状态转移，即去除头尾 i，j 的剩余中间元素
                }
                if(dp[i][j]&&j-i+1>maxlen)
                {
                    maxlen=j-i+1;
                    begin=i;
                }
            }
        }
        return s.substr(begin,maxlen);
    }
};
```