一、题目说明

Longest Palindromic Substring，求字符串中的最长的回文。

Difficuty是Medium

二、我的实现

经过前面4个题目，我对边界考虑越来越"完善"了。

总共提交了5次：

第1、2次：Wrong Answer

主要是**"cbbd"**错误了，重复的判断逻辑上出了点小问题

第3、4次： Time Limit Exceeded

我本地代码运行没问题的，但是提交后，报错。给了一个超长的用例：

```
3210123210012321001232100123210012321001232100123210012321001232100123210
0123210012321001232100123210012321001232100123210012321001232100123210012
3210012321001232100123210012321001232100123210012321001232100123210012321
0012321001232100123210012321001232100123210012321001232100123210012321001
2321001232100123210012321001232100123210012321001232100123210012321001232
1001232100123210012321001232100123210012321001232100123210012321001232100
1232100123210012321001232100123210012321001232100123210012321001232100123
2100123210012321001232100123210012321001232100123210012321001232100123210
0123210012321001232100123210012321001232100123210012321001232100123210012
3210012321001232100123210012321001232100123210012321001232100123210012321
0012321001232100123210012321001232100123210012321001232100123210012321001
2321001232100123210012321001232100123210012321001232100123210012321001232
100123210123210012321001232100123210123
```

我本地运行是ok，但是提交后不行。

我继续优化代码，第5次提交终于对了。本次提交代码性能如下：

```
Runtime: 488 ms, faster than 5.81% of C++ online submissions for Longest
Palindromic Substring.
Memory Usage: 8.7 MB, less than 91.03% of C++ online submissions for Longest
Palindromic Substring.
```

完整代码如下

```
#include<iostream>
#include<string>
using namespace std;

class Solution{
    public:
        string longestPalindrome(string s){
            int maxLength=1;
            string maxStr = s.substr(0,1);
```

```cpp
            int len = s.length();
            if(len<=1){
                return s;
            }else{
                int i = 0;
                while(i<len){
                    int j = len-1;
                    while(j>i){
                        int t=i,k=j;
                        while(t<k && s[t]==s[k]){
                            t++;
                            k--;
                        }
                        if(t==k || t==k+1 && s[t]==s[k]){
                            if(j-i+1>maxLength){
                                maxLength = j-i+1;
                                maxStr = s.substr(i,j-i+1);
                            }
                            break;
                        }
                        j--;
                    }

                    i++;
                }
            }
            return maxStr;
        }
};

int main(){
    Solution s;
    cout<<s.longestPalindrome("ac")<<endl;
    cout<<s.longestPalindrome("babad")<<endl;
    cout<<s.longestPalindrome("cbbd")<<endl;

cout<<s.longestPalindrome("32101232100123210012321001232100123210012321001232100
12321001232100123210012321001232100123210012321001232100123210012321001232100123
21001232100123210012321001232100123210012321001232100123210012321001232100123210
01232100123210012321001232100123210012321001232100123210012321001232100123210012
32100123210012321001232100123210012321001232100123210012321001232100123210012321
00123210012321001232100123210012321001232100123210012321001232100123210012321001
23210012321001232100123210012321001232100123210012321001232100123210012321001232
10012321001232100123210012321001232100123210012321001232100123210012321001232100
12321001232100123210012321001232100123210012321001232100123210012321001232100123
21001232100123210012321001232100123210012321001232100123210012321001232100123210
01232100123210012321001232100123210012321001232100123210012321001232100123210012
32100123210012321001232100123210012321001232100123210012321001232100123210012321
00123210012321001232100123210012321012321001232100123210012321001232100123210123"); 

    return 0;
}
```