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

**Example 1:**

**Input:** "babad"

**Output:** "bab"

**Note:** "aba" is also a valid answer.

**Example 2:**

**Input:** "cbbd"

**Output:** "bb"

题目：

最长的palindrome数

就是agukuga这种

关于中间对称

但是这样里有两种情况，

一种是12321这种，中间就一个数

123321，也是palindrome，中间是两个数

思路，创造一个private方法，用一个pointer找到中间那个轴，然后往两边while loop,只要相等就继续， 然后再Public方法里边不停地遍历这个轴，一直取到最大值，

class Solution {

public String longestPalindrome(String s) {

if(s==null||s.length()<1) return"";

int start=0; //为什么不用max，因为这题让我们return 一个string

int end=0;

for (int i=0;i<s.length();i++){

int len=Math.max(findMid(s,i,i+1),findMid(s,i,i)); //这里很重要，s,i,i+1代表的是abccba型，s,i,i是abcba型

if (len>end-start+1){

start=i-(len-1)/2; 如果是单数，例如

end=i+len/2;

}

}

return s.substring(start,end+1);

}

private static int findMid(String s, int start, int end){

if(end<start||s==null) return 0;

while(end<s.length()&&start>=0&&s.charAt(start)==s.charAt(end)){

start--;

end++;

}

因为==就会c变start—end++， 所以在最后一个循环，我们多弄了一次

abcbadd

到a的时候，start已经是0，end已经是4，

理论上是end-start+1, 减去我们两个多家的，就是end-start-1

}

}