```
/*
给定一个字符串，你的任务是计算这个字符串中有多少个回文子串。
具有不同开始位置或结束位置的子串，即使是由相同的字符组成，也会被计为是不同的子串。
示例 1:
输入: "abc"
输出: 3
解释: 三个回文子串: "a", "b", "c".
示例 2:
输入: "aaa"
输出: 6
说明: 6个回文子串: "a", "a", "a", "aa", "aa", "aaa".
注意:
    输入的字符串长度不会超过1000。
来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/palindromic-substrings
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:

- 方法一:依次枚举长度为1 $length$的子字符串,分别判断是否为回文的，时间复杂度 $T(\frac{n}{2}\frac{n^2+n}{2}) \rightarrow O(n^3)$.
- 其他方法待二刷.

方法一:C++_枚举法

```
class Solution
{
    /* [ left , right ]*/
    private:
        bool isloopstr(string& s , int left , int right)
        {
            bool ret_val = true;
            while(left < right)
            {
                if(s[left] == s[right])
                {
                    left++;
                    right--;
                }
                else
                {
                    ret_val = false;
                    break;
                }
            }
            return ret_val;
        }

    public:
        int countSubstrings(string s)
```

```
        {
            int ret_val =   0           ;
            int length  =   s.size()    ;
            int i       =   0           ;
            int j       =   0           ;
            int left    =   0           ;
            int right   =   0           ;

            ret_val += length;/*长度为1的都是回文串*/
            for(i = 1; i < length ; i ++)
            {
                for(j = 0 ; j < length - i ; j ++)
                {
                    if(isloopstr(s,j,j+i))
                    {
                        ret_val++;
                    }
                }
            }
            return ret_val;
        }
};
/*
执行结果：
通过
显示详情
执行用时 :320 ms, 在所有 cpp 提交中击败了12.40% 的用户
内存消耗 :8.4 MB, 在所有 cpp 提交中击败了93.86%的用户
*/
```

AlimyBreak
2019.11.16