# 题目

给你一个字符串s，请你返回 两个相同字符之间的最长子字符串的长度 ，计算长度时不含这两个字符。如果不存在这样的子字符串，返回-1。

子字符串是字符串中的一个连续字符序列。

示例 1：

输入：s = "aa"

输出：0

解释：最优的子字符串是两个 'a' 之间的空子字符串。

示例 2：

输入：s = "abca"

输出：2

解释：最优的子字符串是 "bc" 。

示例 3：

输入：s = "cbzxy"

输出：-1

解释：s 中不存在出现出现两次的字符，所以返回 -1 。

示例 4：

输入：s = "cabbac"

输出：4

解释：最优的子字符串是"abba"，其他的非最优解包括"bb"和""。

提示：

1 <= s.length <= 300

s 只含小写英文字母

# 分析

class Solution {

public:

int maxLengthBetweenEqualCharacters(string s) {

int first=0; //首指针

int last=s.size()-1; //尾指针

int ans=INT\_MIN; //记录最大值

//通过尾部对首部的匹配，找出相同的两头，计算之间的值

for(int i=first;i<s.size();i++){

//从末尾开始匹配

for(int j=last;j>i;j--){

if(s[i]!=s[j]) continue;

else {

//保存最大值

ans=max(ans,j-i-1);

break;

}

}

}

if(ans>=0) return ans;

else return -1;

}

};

或：

class Solution {

public:

    int maxLengthBetweenEqualCharacters(string s) {

        int ret = INT\_MIN;

        for(int i=0;i<s.size();i++)

        {

            for(int j=s.size()-1;j>i;j--)

            {

                if(s[i]!=s[j])

                {

                    continue;

                }

                else

                {

                    ret =  max(ret,j-i-1);

                    break;

                }

            }

        }

        if(ret>=0)   return ret;

        else return -1;

    }

};