# 题目

给你一个字符串s，它仅包含字符'a'和'b'。

你可以删除s中任意数目的字符，使得s平衡。当不存在下标对(i,j)满足i < j，且s[i] = 'b'的同时s[j]= 'a'，此时认为s是平衡的。

请你返回使s平衡的最少删除次数。

示例 1：

输入：s = "aababbab"

输出：2

解释：你可以选择以下任意一种方案：

下标从0开始，删除第2和第6个字符（"aababbab" -> "aaabbb"），

下标从0开始，删除第3和第6个字符（"aababbab" -> "aabbbb"）。

示例 2：

输入：s = "bbaaaaabb"

输出：2

解释：唯一的最优解是删除最前面两个字符。

提示：

1 <= s.length <= 10^5

s[i]要么是'a'要么是'b'。

# 分析

## 方法一：枚举

思路：

遍历字符串两次：

1、一次遍历统计字符串中'a'的数量，即righta。

2、二次遍历过程中，对于每个字符，如果是'a'，则将righta减一；如果是'b'，则将leftb加一。然后用leftb和righta的和更新结果res，保证删除后的字符串是平衡的。

代码：

class Solution {

public:

int minimumDeletions(string s) {

int leftb = 0, righta = 0;

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

if (s[i] == 'a') {

righta++;

}

}

int res = righta;

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

char c = s[i];

if (c == 'a') {

righta--;

} else {

leftb++;

}

res = min(res, leftb + righta);

}

return res;

}

};