# 题目

给你一个下标从0开始的字符串s，重复执行下述操作 任意 次：

在字符串中选出一个下标i，并使c为字符串下标i处的字符。并在i左侧（如果有）和 右侧（如果有）各删除一个距离i最近的字符c。

请你通过执行上述操作任意次，使s的长度最小化。

返回一个表示最小化字符串的长度的整数。

示例 1：

输入：s = "aaabc"

输出：3

解释：在这个示例中，s 等于 "aaabc" 。我们可以选择位于下标 1 处的字符 'a' 开始。接着删除下标 1 左侧最近的那个 'a'（位于下标 0）以及下标 1 右侧最近的那个 'a'（位于下标 2）。执行操作后，字符串变为 "abc" 。继续对字符串执行任何操作都不会改变其长度。因此，最小化字符串的长度是 3 。

示例 2：

输入：s = "cbbd"

输出：3

解释：我们可以选择位于下标 1 处的字符 'b' 开始。下标 1 左侧不存在字符 'b' ，但右侧存在一个字符 'b'（位于下标 2），所以会删除位于下标 2 的字符 'b' 。执行操作后，字符串变为 "cbd" 。继续对字符串执行任何操作都不会改变其长度。因此，最小化字符串的长度是 3 。

示例 3：

输入：s = "dddaaa"

输出：2

解释：我们可以选择位于下标 1 处的字符 'd' 开始。接着删除下标 1 左侧最近的那个 'd'（位于下标 0）以及下标 1 右侧最近的那个 'd'（位于下标 2）。执行操作后，字符串变为 "daaa" 。继续对新字符串执行操作，可以选择位于下标 2 的字符 'a' 。接着删除下标 2 左侧最近的那个 'a'（位于下标 1）以及下标 2 右侧最近的那个 'a'（位于下标 3）。执行操作后，字符串变为 "da" 。继续对字符串执行任何操作都不会改变其长度。因此，最小化字符串的长度是 2 。

提示：

1 <= s.length <= 100

s 仅由小写英文字母组成

# 分析

## 方法一：模拟

思路：

使用一个整数mask来表示字符串中出现过的字符。对于字符串中的每个字符，我们将其转换为对应的二进制位，并将该位设置为1，以便后续检查是否出现过相同的字符。最后，我们统计mask中1的个数，即为最小化后的字符串长度。

具体步骤如下：

1、遍历字符串s中的每个字符，计算其相对于'a'的偏移量，并将该偏移量对应的二进制位设置为1，即mask |= 1 << (c - 'a')。

2、最后，统计mask中1的个数，即为最小化后的字符串长度。

这样做的原因是，如果字符串中存在相同的字符，那么它们在mask中对应的二进制位会相同，因此最终mask中1的个数就代表了最终字符串的长度。

代码：

class Solution {

public:

int minimizedStringLength(string s) {

int mask = 0;

for (char c: s) mask |= 1 << (c - 'a');

return countOnes(mask);

}

private:

int countOnes(int n) {

int count = 0;

while (n) {

n = n & (n - 1);

count++;

}

return count;

}

};