# 题目

给你一个括号字符串 s ，它只包含字符 '(' 和 ')' 。一个括号字符串被称为平衡的当它满足：

任何左括号'('必须对应两个连续的右括号'))'。

左括号'('必须在对应的连续两个右括号'))'之前。

比方说"())"，"())(())))"和"(())())))"都是平衡的，")()"，"()))"和"(()))"都是不平衡的。

你可以在任意位置插入字符 '(' 和 ')' 使字符串平衡。

请你返回让 s 平衡的最少插入次数。

示例 1：

输入：s = "(()))"

输出：1

解释：第二个左括号有与之匹配的两个右括号，但是第一个左括号只有一个右括号。我们需要在字符串结尾额外增加一个 ')' 使字符串变成平衡字符串 "(())))" 。

示例 2：

输入：s = "())"

输出：0

解释：字符串已经平衡了。

示例 3：

输入：s = "))())("

输出：3

解释：添加 '(' 去匹配最开头的 '))' ，然后添加 '))' 去匹配最后一个 '(' 。

示例 4：

输入：s = "(((((("

输出：12

解释：添加 12 个 ')' 得到平衡字符串。

示例 5：

输入：s = ")))))))"

输出：5

解释：在字符串开头添加 4 个 '(' 并在结尾添加 1 个 ')' ，字符串变成平衡字符串 "(((())))))))" 。

提示：

1 <= s.length <= 10^5

s 只包含 '(' 和 ')' 。

# 分析

## 方法一：贪心

思路：

遍历字符串s，用leftCount记录当前未匹配的左括号数量，用insertions记录需要插入的括号数量。具体步骤如下：

1、初始化insertions为0，leftCount为0，index为0。

2、循环遍历字符串s，直到index达到字符串长度。

3、对于当前字符c，如果是左括号'('，则leftCount加1，index加1。

4、对于当前字符c，如果是右括号')'，则判断是否存在匹配的左括号：

- 如果leftCount大于0，表示存在未匹配的左括号，将leftCount减1。

- 如果leftCount等于0，表示没有匹配的左括号，需要插入一个左括号，insertions加1。

- 同时，判断下一个字符是否为右括号')'，如果是，表示存在连续的右括号，index加2；如果不是，表示需要插入一个右括号，insertions加1，index加1。

5、循环结束后，将未匹配的左括号数量乘以2，加到insertions中，即为最终需要插入的括号数量。

6、返回insertions作为结果。

代码：

class Solution {

public:

int minInsertions(string s) {

int insertions = 0;

int leftCount = 0;

int length = s.size();

int index = 0;

while (index < length) {

char c = s[index];

if (c == '(') {

leftCount++;

index++;

} else {

if (leftCount > 0) {

leftCount--;

} else {

insertions++;

}

if (index < length - 1 && s[index + 1] == ')') {

index += 2;

} else {

insertions++;

index++;

}

}

}

insertions += leftCount \* 2;

return insertions;

}

};