# 题目

一个括号字符串是只由'(' 和 ')'组成的非空字符串。如果一个字符串满足下面任意一个条件，那么它就是有效的：

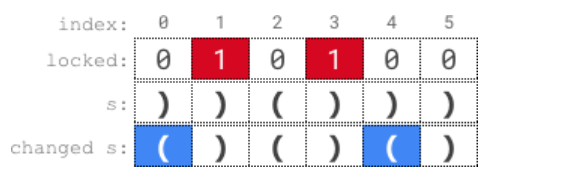
* 字符串为().
* 它可以表示为AB（A与B连接），其中A和B都是有效括号字符串。
* 它可以表示为(A)，其中 A 是一个有效括号字符串。

给你一个括号字符串s和一个字符串locked，两者长度都为n。locked是一个二进制字符串，只包含'0'和'1'。对于locked中每一个下标i：

* 如果locked[i]是'1' ，你不能改变s[i]。
* 如果locked[i]是'0'，你可以将s[i]变为'('或者')'。

如果你可以将s变为有效括号字符串，请你返回true，否则返回false。

示例 1：



输入：s = "))()))", locked = "010100"

输出：true

解释：locked[1] == '1' 和 locked[3] == '1' ，所以我们无法改变 s[1] 或者 s[3] 。

我们可以将 s[0] 和 s[4] 变为 '(' ，不改变 s[2] 和 s[5] ，使 s 变为有效字符串。

示例 2：

输入：s = "()()", locked = "0000"

输出：true

解释：我们不需要做任何改变，因为 s 已经是有效字符串了。

示例 3：

输入：s = ")", locked = "0"

输出：false

解释：locked 允许改变 s[0] 。

但无论将 s[0] 变为 '(' 或者 ')' 都无法使 s 变为有效字符串。

提示：

n == s.length == locked.length

1 <= n <= 105

s[i] 要么是 '(' 要么是 ')' 。

locked[i] 要么是 '0' 要么是 '1' 。

# 分析

## 方法一：栈

思路：

代码：

class Solution {

public:

bool canBeValid(string s, string locked) {

int n = s.size();

int mx = 0; // 可以达到的最大分数

int mn = 0; // 可以达到的最小分数 与 最小有效前缀对应分数 的较大值

for (int i = 0; i < n; ++i) {

if (locked[i] == '1') {

// 此时对应字符无法更改

int diff;

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

diff = 1;

} else {

diff = -1;

}

mx += diff;

mn = max(mn + diff, (i + 1) % 2);

} else {

// 此时对应字符可以更改

++mx;

mn = max(mn - 1, (i + 1) % 2);

}

if (mx < mn) {

// 此时该前缀无法变为有效前缀

return false;

}

}

// 最终确定 s 能否通过变换使得分数为 0（成为有效字符串）

return mn == 0;

}

};