# 题目

给你一个只包含三种字符的字符串，支持的字符类型分别是 '('、')' 和 '\*'。请你检验这个字符串是否为有效字符串，如果是有效字符串返回 true 。

有效字符串符合如下规则：

任何左括号 '(' 必须有相应的右括号 ')'。

任何右括号 ')' 必须有相应的左括号 '(' 。

左括号 '(' 必须在对应的右括号之前 ')'。

'\*' 可以被视为单个右括号 ')' ，或单个左括号 '(' ，或一个空字符串。

一个空字符串也被视为有效字符串。

示例 1：

输入：s = "()"

输出：true

示例 2：

输入：s = "(\*)"

输出：true

示例 3：

输入：s = "(\*))"

输出：true

提示：

1 <= s.length <= 100

s[i] 为 '('、')' 或 '\*'

# 分析

## 方法一：栈

class Solution {

public:

bool checkValidString(string s) {

stack<int> left;

stack<int> star;

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

char c = s[i];

//记录'('的位置

if (c == '(') left.push(i);

// 记录'\*'的位置

if (c == '\*') star.push(i);

if (c == ')') {

// 如果存在')'，则判断是否存在'('，有则弹出

if (!left.empty()) {

left.pop();

// 如果存在'\*'，则判断是否存在'('，有则弹出

} else if (!star.empty()) {

star.pop();

} else {

return false;

}

}

}

// 判断最后'('和'\*'的数量是否匹配

while (!left.empty()) {

int leftPos = left.top();

if (star.empty()) return false;

int starPos = star.top();

if (starPos > leftPos) {

left.pop();

star.pop();

} else {

return false;

}

}

return true;

}

};

## 方法二：动态规划