**[Valid Parenthesis String](https://leetcode.com/problems/valid-parenthesis-string/)**

Given a string s containing only three types of characters: '(', ')' and '\*', return true *if* s *is****valid***.

The following rules define a **valid** string:

* Any left parenthesis '(' must have a corresponding right parenthesis ')'.
* Any right parenthesis ')' must have a corresponding left parenthesis '('.
* Left parenthesis '(' must go before the corresponding right parenthesis ')'.
* '\*' could be treated as a single right parenthesis ')' or a single left parenthesis '(' or an empty string "".

**Example 1:**

**Input:** s = "()"

**Output:** true

**Example 2:**

**Input:** s = "(\*)"

**Output:** true

**Example 3:**

**Input:** s = "(\*))"

**Output:** true

**Constraints:**

* 1 <= s.length <= 100
* s[i] is '(', ')' or '\*'.

class Solution {

public:

bool checkValidString(string s) {

int leftMin = 0, leftMax = 0;

for (char c : s) {

if (c == '(') {

leftMin++;

leftMax++;

} else if (c == ')') {

leftMin--;

leftMax--;

} else {

leftMin--;

leftMax++;

}

if (leftMax < 0) return false;

if (leftMin < 0) leftMin = 0;

}

return leftMin == 0;

}

};

Link : <https://leetcode.com/problems/valid-parenthesis-string/submissions/1226058077/?envType=daily-question&envId=2024-04-07>