**Valid Parenthesis String**

Given a string containing only three types of characters: '(', ')' and '\*', write a function to check whether this string is valid. We define the validity of a string by these rules:

1. Any left parenthesis '(' must have a corresponding right parenthesis ')'.
2. Any right parenthesis ')' must have a corresponding left parenthesis '('.
3. Left parenthesis '(' must go before the corresponding right parenthesis ')'.
4. '\*' could be treated as a single right parenthesis ')' or a single left parenthesis '(' or an empty string.
5. An empty string is also valid.

class Solution **{**

public boolean checkValidString**(**String s**)** **{**

Stack**<**Character**>** generalStack **=** **new** Stack**<**Character**>();**

int auxStars**=**0**;**

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

**if(**s**.**charAt**(**i**)==**'('**){**

generalStack**.**push**(**'('**);**

**}else** **if(**s**.**charAt**(**i**)==**'\*'**){**

**if(!**generalStack**.**isEmpty**()** **&&** generalStack**.**peek**()==**'('**){**

generalStack**.**pop**();**

auxStars**+=**2**;**

**}else{**

generalStack**.**push**(**'\*'**);**

**}**

**}else** **if(**s**.**charAt**(**i**)==**')'**){**

**if(!**generalStack**.**isEmpty**()){**

generalStack**.**pop**();**

**}else** **if(**auxStars**!=**0**){**

auxStars**--;**

**}else{**

**return** **false;**

**}**

**}**

**}**

**if(**generalStack**.**isEmpty**()||**generalStack**.**peek**()** **==** '\*'**)**

**return** **true;**

**return** **false;**

**}**

**}**