

```

1  /*
2  给定一个由 '(' 和 ')' 括号组成的字符串 S，我们需要添加最少的括号（ '(' 或是 ')'，可以在任何位置），以使得到的括号字符串有效。
3
4  从形式上讲，只有满足下面几点之一，括号字符串才是有效的：
5
6      它是一个空字符串，或者
7      它可以被写成 AB（A 与 B 连接），其中 A 和 B 都是有效字符串，或者
8      它可以被写作 (A)，其中 A 是有效字符串。
9
10 给定一个括号字符串，返回为使结果字符串有效而必须添加的最少括号数。
11
12
13
14 示例 1：
15
16 输入："())"
17 输出：1
18
19 示例 2：
20
21 输入："((("
22 输出：3
23
24 示例 3：
25
26 输入："()"
27 输出：0
28
29 示例 4：
30
31 输入："()())(("
32 输出：4
33
34
35
36 提示：
37
38     S.length <= 1000
39     S 只包含 '(' 和 ')' 字符。
40
41
42
43 来源：力扣（LeetCode）
44 链接：https://leetcode-cn.com/problems/minimum-add-to-make-parentheses-valid
45 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
46 */

```

分析:

- 利用栈(计数器)做抵消
 - 遍历字符串所有元素.遇到'('或')'则对应栈压入一个当前字符(对应计数器加1);

- 在遍历过程中,在遇'('的情况下,如果对应')'里面的存量不为0,就把两个栈都remove他们的栈顶元素(两种计数都减去1);
- 最后将两个栈的剩余元素个数总和(两个计数器之和)返回.

方法一:C++_栈(计数)抵消法

```
1  class Solution
2  {
3      public:
4          int minAddToMakeValid(string S)
5          {
6              int left    = 0;
7              int right   = 0;
8
9              for(int i = 0 ; i < S.size() ; i++)
10             {
11                 if(S[i] == '(')
12                 {
13                     left++;
14                 }
15                 else
16                 {
17                     right++;
18                     if(left>0)
19                     {
20                         left--;
21                         right--;
22                     }
23                 }
24             }
25
26             return left+right;
27
28         }
29     };
30
31     /*
32     执行结果:
33     通过
34     显示详情
35     执行用时 :0 ms, 在所有 cpp 提交中击败了100.00% 的用户
36     内存消耗 :8.5 MB, 在所有 cpp 提交中击败了85.63%的用户
37     */
```