921. Minimum Add to Make Parentheses Valid

Solved ✓

Medium   ◇ Topics   🔒 Companies

A parentheses string is valid if and only if:

- It is the empty string,
- It can be written as `AB` (`A` concatenated with `B`), where `A` and `B` are valid strings, or
- It can be written as `(A)`, where `A` is a valid string.

You are given a parentheses string `s`. In one move, you can insert a parenthesis at any position of the string.

- For example, if `s = "())) "`, you can insert an opening parenthesis to be `"(()) "` or a closing parenthesis to be `"())) "`.

Return *the minimum number of moves required to make* `s` *valid*.

Example 1:

```
Input: s = "())"
Output: 1
```

Example 2:

```
Input: s = "((("
Output: 3
```

Constraints:

- `1 <= s.length <= 1000`
- `s[i]` is either `'('` or `')'`.

Approach 1:

```cpp
class Solution {
public:
    int minAddToMakeValid(string s) {
        map<char,int> m;
        m['(']=0;
        int count=0;

        for(auto i:s){
            if(i=='(') m['(']++;
            else{
                if(m['(']==0) count++;
                else m['(']--;
            }
        }

        return count+m['('];
    }
};
```

$$T(n) : O(n)$$
$$S(n) : O(1)$$

we can also use stack to do the task

we can also use stack to do the task