Longest Valid Parentheses

We must find the length of the longest substring where parentheses are properly matched and well formed.

Stack Approach (O(n) Time, O(n) Space):

- Push indices of '(' onto stack.
- When encountering ')':
  - Pop stack (if possible) → calculate length of valid substring.
  - Maintain a base index for last unmatched ')'.

Two-Pass Counter Approach (O(n) Time, O(1) Space):

- Use two passes:
  1. Left to right:
     - Count left and right parentheses:
     - If left == right, update maxLen.
     - If right > left, reset counters.
  2. Right to left:
     - Same logic but swap roles of left and right.

Why Two Passes?

- Left to right ensures no surplus ')' on left.
- Right to left ensures no surplus '(' on right.
- Combined, they cover all valid substrings.

Complexity:

- Time: O(n)
- Space: O(1)