1190. Reverse Substrings Between Each Pair of Parentheses

You are given a string s that consists of lower case English letters and brackets.

Reverse the strings in each pair of matching parentheses, starting from the innermost one.

Your result should not contain any brackets.

Example 1:

```
Input: s = "(u(love)i)"
Output: "iloveu"
Explanation: The substring "love" is reversed first,
then the whole string is reversed.
```

```
class Solution {
    public String reverseParentheses(String s) {

        // stack with two passes (O(N) time, space)
        int n = s.length();
        Stack<Integer> stack = new Stack<>();
        int[] pair = new int[n];

        // first pass
        for (int i = 0; i < n; i++) {
            if (s.charAt(i) == '(') {
                stack.push(i);
            }

            if (s.charAt(i) == ')') {
                int j = stack.pop();
                pair[i] = j;
                pair[j] = i;
            }
        }

        // second pass
        StringBuilder sb = new StringBuilder();
        for (int i = 0, d = 1; i < n; i += d) {
            if (s.charAt(i) == '(' || s.charAt(i) == ')') {
                i = pair[i];
                d = -d;
            } else {
                sb.append(s.charAt(i));
            }
        }

        return sb.toString();
    }
}
```