

1190 反转没对括号间的子串

Label: 栈

给出一个字符串 `s`（仅含有小写英文字母和括号）。

请你按照从括号内到外的顺序，逐层反转每对匹配括号中的字符串，并返回最终的结果。

注意，您的结果中 不应 包含任何括号。

输入: `s = "(abcd)"`

输出: `"dcba"`

输入: `s = "(u(love)i)"`

输出: `"iloveu"`

输入: `s = "(ed(et(oc))el)"`

输出: `"leetcode"`

输入: `s = "a(bcdefghijkl(mno)p)q"`

输出: `"apmno1kjihgfedcbq"`

- 栈 + 队列

```
class Solution {
    public String reverseParentheses(String s) {
        Stack<Character> stack = new Stack<>();
        Queue<Character> queue = new LinkedList<>();
        for (int i = 0; i < s.length(); i++) {
            if (s.charAt(i) == ')') {
                while (!stack.isEmpty() && stack.peek() != '(') {
                    queue.offer(stack.pop());
                }
                if (!stack.isEmpty()) stack.pop(); // 弹出 (
                // 反向入栈
                while (!queue.isEmpty()) {
                    stack.add(queue.poll());
                }
            } else { // 入栈
                stack.add(s.charAt(i));
            }
        }
        // 形成字符串
        StringBuffer sb = new StringBuffer();
        while (!stack.isEmpty()) {
            sb.append(stack.pop());
        }
        return sb.reverse().toString();
    }
}
```

- 双指针

```
class Solution {
    public String reverseParentheses(String s) {
        char[] arr = s.toCharArray();
        int right = 0, left, len = arr.length;
        while(right < len){
            if(arr[right] != ')'){
                right += 1;
            }
            // 找打右括号, 从该点向前遍历
            else{
                left = right;
                while(arr[left] != '('){
                    left -= 1;
                }
                arr[left] = '0'; // 考虑用 0 来占位
                arr[right] = '0';
                reverse(arr, left, right);
            }
        }

        StringBuffer ans = new StringBuffer();
        for(int i = 0; i < len; i++){
            if(arr[i] != '0'){
                ans.append(arr[i]);
            }
        }
        return ans.toString();
    }

    public void reverse(char[] arr, int start, int end){
        while(start < end){
            char temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }
    }
}
```