

Name:

Kun Liu

No.:

MARCH TWENTY FOURTH, 2026

Class:

CS: Data Structure
Homework 2nd: Stack and Queue

1. 回文是指正读反读均相同的字符序列，如“abba”和“abdba”均是回文，但“good”不是回文。试写一个算法判定给定的字符向量是否为回文。(提示：将一半字符入栈)
2. 设从键盘输入一整数的序列： $a_1, a_2, a_3, \dots, a_n$ ，试编写算法实现：用栈结构存储输入的整数，当 $a_i \neq -1$ 时，将 a_i 进栈；当 $a_i = -1$ 时，输出栈顶整数并出栈。算法应对异常情况（入栈满等）给出相应的信息。
3. 从键盘上输入一个后缀表达式，试编写算法计算表达式的值。规定：逆波兰表达式的长度不超过一行，以 \$ 符作为输入结束，操作数之间用空格分隔，操作符只可能有 +、-、*、/ 四种运算。例如：234 34+2*\$。
4. 如果允许在循环队列的两端都可以进行插入和删除操作。要求：①写出循环队列的类型定义；②写出“从队尾删除”和“从队头插入”的算法。
5. 已知Ackermann函数定义如下：

$$\text{Ack}(x) = \begin{cases} n + 1, & \text{当 } m = 0 \text{ 时} \\ \text{Ack}(m - 1, 1), & \text{当 } m \neq 0, n = 0 \text{ 时} \\ \text{Ack}(m - 1, \text{Ack}(m, n - 1)), & \text{当 } m \neq 0, n \neq 0 \text{ 时} \end{cases}$$

- ①写出计算Ack(m,n)的递归算法，并根据此算法给出Ack(2,1)的计算过程。
- ②写出计算Ack(m,n)的非递归算法。