

栈和队列

(仅供课后复习使用, 不得外传)

一、填空题

1. 栈和队列都是 线性 结构, 对于栈只能在 栈顶 插入和删除元素; 对于队列只能在 队尾 插入和 队头(队首) 删除元素。

2. 栈是一种特殊的线性表, 允许插入和删除运算的一端称为 栈顶。不允许插入和删除运算的一端称为 栈底。

3. 是被限定为只能在线性表的一端进行插入运算, 在表的另一端进行删除运算的 队列 线性表。

4. 向栈中压入元素的操作是先 移动栈顶指针, 后 压入(存入, 插入)元素。

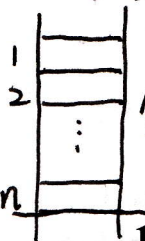
5. 从循环队列中删除一个元素时, 其操作是先 移动队首指针, 后 读(取出)元素。

6. 顺序 队列有假上溢现象, 产生此现象的原因是 队头、队尾, 采用 循环 队列可消除此现象。
指针只增不减

7. 顺序栈无假上溢现象的原因是 栈顶指针随着入栈、出栈操作有增有减, 不会出现只增不减现象; 链栈不设头结点的原因是 因链栈只在头指针所指的位置操作, 故无须设置头指针; 结点。

8. 设长度为 n 的 链队列 用单循环链表表示, 若只设 头指针, 则出队和入队操作的时间复杂度分别为 $O(1)$, $O(n)$; 若只设 尾指针, 则出队和入队操作的时间复杂度分别为 $O(1)$, $O(1)$ 。

9. 在一个具有 n 个单元的顺序栈中, 假定以地址高端 (即下标为 n 的单元) 作为栈底, 以 top 作为栈顶指针, 则当向栈中压入一个元素时, top 指针的变化是 $top = top - 1$ 。



10. 多个栈共存时, 最好用 链式存储结构 作为存储结构。

11. 有 5 个数据依次入栈: 1, 2, 3, 4, 5。在各种出栈序列中, 以 3, 4 先出栈的序列有

(3在4之前出栈): 3个分别是 34521, 34215, 34251

12. 已知链队列的头尾指针分别是 f 和 r, 则将值 x 入队的操作序列是:

$S = (\text{listNode}^*) \text{malloc} (\text{sizeof} (\text{list node}));$

二、判断 (判断下列概念的正确性, 并作出简要的说明。)

✖ (X) 线性表的每个结点只能是一个简单类型, 而链表的每个结点可以是一个复杂类型。逻辑结构概念与之素数据类型无关。
 $S \rightarrow \text{data} = x; S \rightarrow \text{next} = r \rightarrow \text{next};$
 $r \rightarrow \text{next} = S;$
 $r = S;$


✖ (✓) 栈是一种对所有插入、删除操作限于在表的一端进行的线性表, 是一种后进先出型结构。

✖ (✓) 对于不同的使用者, 一个表结构既可以是栈, 也可以是队列, 也可以是线性表。

✖ (X) 栈和链表是两种不同的数据结构。栈是一种运受限的特殊线性表。

✖ (X) 栈和队列是一种非线性数据结构。线性。非线性结构包括树和图。

✖ (✓) 栈和队列的存储方式既可是顺序方式, 也可是链接方式。

✖ (✓) 两个栈共享一片连续内存空间时, 为提高内存利用率, 减少溢出机会, 应把两个栈的栈底分别设在这片内存空间的两端。


✖ (X) 队是一种插入与删除操作分别在表的两端进行的线性表, 是一种先进后出型结构。

✖ (X) 一个栈的输入序列是 12345, 则栈的输出序列不可能是 12345。明显矛盾。
1进1出, 2进2出, ...

✖ (✓) 栈和队列的存储方式, 既可以是顺序方式, 也可以是链式方式。5进5出。

三、选择题

■ 1. 栈中元素的进出原则是

(B).

A. 先进先出 B. 后进先出 C. 栈空则进 D. 栈满则出

✖ ■ 2. 若已知一个栈的入栈序列是 1, 2, 3, ..., n, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$, 若 $p_1 = n$, 则 p_i 为:

pn, 若 $p_1 = n$, 则 p_i 为:

(C).

A. i B. n-i C. n-i+1 D. 不确定

当 $p_1 = n$ 时, 表示 n 是最先出栈的, 根据栈的 LIFO 规则, n 必是最后入栈的。

所以输入顺序: 1, 2, 3, ..., n
出栈序列: n, n-1, ..., n-i+1, ..., 3, 2, 1

■ 3. 一个递归算法必须包括 () B.

A. 递归部分 B. 终止条件和递归部分 C. 迭代部分 D. 终止条件和迭代部分

■ 4. 栈一般在 () 中应用 D.

A. 递归调用 B. 子程序调用 C. 表达式求值 D. A, B, C 皆是

★ ■ 5. 一个栈的入栈序列为 a, b, c, d, e, 则栈的不可能出栈序列是 () D.

A. abcde B. edcba C. decba D. dceab

e 是最后一个元素

故 e 之后的元素不可能是递增序列

■ 6. 判定一个栈 ST (最多元素为 m0) 为空的条件是

(B)

A. $ST \rightarrow top \neq 0$ B. $ST \rightarrow top = 0$ C. $ST \rightarrow top \neq m0$ D. $ST \rightarrow top = m0$

★ ■ 7. 判定一个队列 QU (最多元素为 m0) 为满队列的条件是

AB.

A. $QU \rightarrow rear - QU \rightarrow front == m0$ B. $QU \rightarrow front == (QU \rightarrow rear + 1) \% m0$

C. $QU \rightarrow front == QU \rightarrow rear$ D. $QU \rightarrow front == QU \rightarrow rear + 1$

★ ■ 8. 设存放在 $A[1..n-1]$ 中的循环队列, 用 front 和 length 分别表示实际队头位置和

①.

队列长度, 则队尾元素的实际位置是:

D.

A. $front + length - 1$; B. $(front + length) \bmod n$

C. $front + length$ D. $(front + length - 1) \bmod n$

■ 9. 用单循环链表表示队列, 正确的说法是:

A. 可设一个头指针使入队、出队都方便;

B. 可设一个尾指针使入队、出队都方便;

C. 必须设头指针才能使入队、出队都方便;

D. 无论如何, 只可能使入队方便;

B. 注意: 这里题干指的是“单循环链表”

方便是指操作时间为 $O(1)$

设头尾指针肯定会使入队、出队方便, 但这不是“必须”的。

因为只设一个尾指针也可使入队、出队都方便。

四、简答题

1. 如何用两个栈来实现一个队列? 简述算法思想并给出算法。

设两栈 S1 和 S2 分别用作输入和输出, 入队操作解释为 $Push(S1, x)$, 出队操作为 $Pop(S2)$, 若 S2 为空, 则将 S1 中之元素依次出栈, 同时将其压入栈 S2, 待 S1 中所有之元素出栈并进入栈 S2 后, 令 S2 出栈。若 S2 非空, 则直接在 S2 上做一次出栈动作。

△ 2. 说明线性表、栈与队的异同点。

相同点：都是线性结构。（逻辑结构相同）
相异点：运算规则有区别

在上述基础上展开。

3. 设有编号为 1, 2, 3, 4 的四辆列车，顺序进入一个栈式结构的车站，具体写出这四辆列车开出车站的所有可能的顺序。

全进之后再出：4, 3, 2, 1

进3个之后再出：3, 4, 2, 1; 3, 2, 4, 1; 3, 2, 1, 4

进2个之后再出：2, 4, 3, 1; 2, 3, 4, 1; 2, 1, 3, 4; 2, 1, 4, 3; 2, 3, 1, 4

进1个之后再出：1, 4, 3, 2; 1, 3, 2, 4; 1, 3, 4, 2; 1, 2, 3, 4; 1, 2, 4, 3;

4. 假设正读和反读都相同的字符序列为“回文”，例如，‘abba’和‘abcba’是回文，

‘abcde’和‘ababab’则不是回文。试写一算法判别读入的字符序列是否为回文？

（可以用伪代码描述算法实现思想）

思想：可将输入字符序列放在一个字符数组中；然后将数组中字符依次入栈，再将数组中的字符和依次出栈字符进行比较。

若未发现不匹配字符，则字符串为回文。

5. 顺序队的“假溢出”是怎样产生的？如何知道循环队列是空还是满？

一般的一维数组当队列的尾指针（rear）到了数组上界时，就不能再进行入队列操作了，但其实该一维数组中可能还存在空闲空间，我们称这种情形为假溢出。采用循环队列可解决该问题，3种判定方法（见课件）

△ 6. 设循环队列的容量为 40（序号从 0 到 39），现经过一系列的入队和出队运算后，有

① front=11, rear=19; ② front=19, rear=11; 问在这两种情况下，循环队列中各有元素多少个？

队列长度计算公式：
$$\begin{cases} \text{当 } rear \geq front \text{ 时, 队列长度 } L = rear - front \\ \text{当 } rear < front \text{ 时, } L = m + (rear - front) \end{cases}$$

统一公式：
$$L = (m + (rear - front)) \% m$$
（其中 m 为数组大小）

故：

7. 补充一题。请用伪代码说明如何利用两个栈共享一段存储空间。

① 8

② $(40 + 11 - 19)$

$\% 40 = 32$