
注意:

1. 请独立完成, 杜绝抄袭;
2. 提交时间: 10-15。

第3章 栈和队列

1. 选择题

- (1) 若让元素 1, 2, 3, 4, 5 依次进栈, 则出栈次序不可能出现在 () 种情况。
A. 5, 4, 3, 2, 1 B. 2, 1, 5, 4, 3 C. 4, 3, 1, 2, 5 D. 2, 3, 5, 4, 1
- (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 ()。
A. i B. n-i C. n-i+1 D. 不确定
- (3) 数组 $Q[n]$ 用来表示一个循环队列, f 为当前队列头元素的前一位置, r 为队尾元素的位置, 假定队列中元素的个数小于 n , 计算队列中元素个数的公式为 ()。
A. $r-f$ B. $(n+f-r)\%n$ C. $n+r-f$ D. $(n+r-f)\%n$
- (4) 链式栈结点为: (data,link), top 指向栈顶. 若想摘除栈顶结点, 并将删除结点的值保存到 x 中, 则应执行操作 ()。
A. $x=top->data; top=top->link;$ B. $top=top->link; x=top->link;$
C. $x=top; top=top->link;$ D. $x=top->link;$
- (5) 设有一个递归算法如下
int fact(int n) { //n 大于等于 0
 if(n<=0) return 1;
 else return n*fact(n-1); }
则计算 fact(n) 需要调用该函数的次数为 ()。
A. n+1 B. n-1 C. n D. n+2
- (6) 栈在 () 中有所应用。
A. 递归调用 B. 函数调用 C. 表达式求值 D. 前三个选项都有

(7) 为解决计算机主机与打印机间速度不匹配问题, 通常设一个打印数据缓冲区。主机将要输出的数据依次写入该缓冲区, 而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是 ()。

- A. 队列 B. 栈 C. 线性表 D. 有序表

(8) 设栈 S 和队列 Q 的初始状态为空, 元素 e1、e2、e3、e4、e5 和 e6 依次进入栈 S, 一个元素出栈后即进入 Q, 若 6 个元素出队的序列是 e2、e4、e3、e6、e5 和 e1, 则栈 S 的容量至少应该是 ()。

- A. 2 B. 3 C. 4 D. 6

(9) 若一个栈以向量 V[1..n] 存储, 初始栈顶指针 top 设为 n+1, 则元素 x 进栈的正确操作是 ()。

- A. $\text{top}++$; $V[\text{top}]=x$; B. $V[\text{top}]=x$; $\text{top}++$;
C. $\text{top}-$; $V[\text{top}]=x$; D. $V[\text{top}]=x$; $\text{top}-$;

(10) 设计一个判别表达式中左, 右括号是否配对出现的算法, 采用 () 数据结构最佳。

- A. 线性表的顺序存储结构 B. 队列
C. 线性表的链式存储结构 D. 栈

(11) 用链接方式存储的队列, 在进行删除运算时 ()。

- A. 仅修改头指针 B. 仅修改尾指针
C. 头、尾指针都要修改 D. 头、尾指针可能都要修改

(12) 循环队列存储在数组 A[0..m] 中, 则入队时的操作为 ()。

- A. $\text{rear}=\text{rear}+1$ B. $\text{rear}=(\text{rear}+1)\%(m-1)$
C. $\text{rear}=(\text{rear}+1)\%m$ D. $\text{rear}=(\text{rear}+1)\%(m+1)$

(13) 最大容量为 n 的循环队列, 队尾指针是 rear, 队头是 front, 则队空的条件是 ()。

- A. $(\text{rear}+1)\%n==\text{front}$ B. $\text{rear}==\text{front}$
C. $\text{rear}+1==\text{front}$ D. $(\text{rear}-1)\%n==\text{front}$

(14) 栈和队列的共同点是 ()。

- A. 都是先进先出 B. 都是先进后出
C. 只允许在端点处插入和删除元素 D. 没有共同点

(15) 一个递归算法必须包括 ()。

- 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

2. 算法设计题 (请进行算法分析, 并写出相应的函数代码. 最后两题为选做题)

(1) 将编号为 0 和 1 的两个栈存放于一个数组空间 V[m] 中, 栈底分别处于数组的两端。当第 0 号栈的栈顶指针 top[0] 等于 -1 时该栈为空, 当第 1 号栈的栈顶指针 top[1] 等于 m 时该栈为空。

两个栈均从两端向中间增长。试编写双栈初始化, 判断栈空、栈满、进栈和出栈等算法的函数。
双栈数据结构的定义如下:

```
Typedef struct
{int top[2],bot[2];           //栈顶和栈底指针
  SElemType *V;               //栈数组
  int m;                       //栈最大可容纳元素个数
}DblStack
```

(2) 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)

(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) 假设以带头结点的循环链表表示队列, 并且只设一个指针指向队尾元素站点(注意不设头指针), 试编写相应的置空队、判队空、入队和出队等算法。

(5) 假设以数组 $Q[m]$ 存放循环队列中的元素, 同时设置一个标志 tag , 以 $tag == 0$ 和 $tag == 1$ 来区别在队头指针($front$)和队尾指针($rear$)相等时, 队列状态为“空”还是“满”。试编写与此结构相应的插入($enqueue$)和删除($dlqueue$)算法。

(6) 已知 Ackermann 函数定义如下:

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

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。