

数据结构第四章

王天一 320200931301

1. 简要回答解释术语

1. 栈: 栈是一个下限为常数, 上限为可变化的向量 (或者反之), 栈也称为堆栈或堆阵。
2. 队列: 队列是一个下限和上线只能增加而不能减少的向量。
3. 假溢出: 由于空间分配或管理不当, 导致的空间还有, 数据却无法装入的现象。
4. 递归函数: 一个过程函数的定义中又应用了它自身, 那么该函数为递归的。

2. 栈与线性表的异同

(1) 栈与线性表的异同

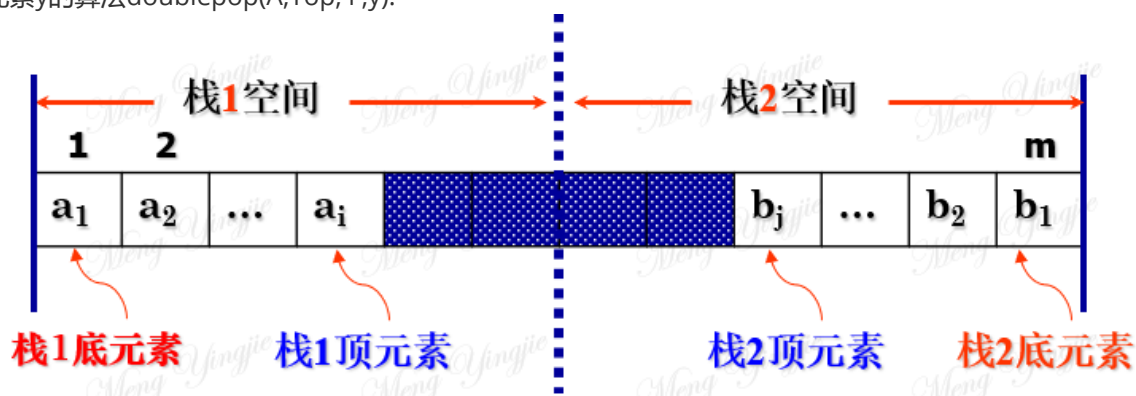
1. 数据对象: 栈和线性表的数据对象都是相同类型的数据放在一起。
2. 数据逻辑关系: 栈和线性表的数据元素间的逻辑关系都是线性关系。
3. 数据运算关系: 线性表的插入和删除可以在表的任意位置进行, 而栈仅能在表的一端进行, 栈中的数据元素的组织是按“后进先出”的原则组织的。

(2) 队列与线性表的异同

1. 数据对象: 队列和线性表的数据对象都是相同类型的数据放在一起。
2. 数据对象逻辑关系: 队列和线性表的数据元素间的逻辑关系。
3. 数据运算关系: 线性表的插入和删除可以在表的任意位置进行, 而队列仅能在表的端点进行, 队列中数据元素的组织是按“先进先出”的原则组织的。

3. 双栈共存问题的算法设计

利用一维数组 $A[1..m]$ 来组织两个栈如图所示, $Top[i]$ 为栈顶指针 (i 为 1 或 2, 分别表示栈 1 或栈 2), 当全部 m 个单元全满时才产生溢出。写出将数据元素 x 压入第 i 个栈的算法 $Doublepush(A, Top, i, x)$; 从第 i 个栈弹出元素 y 的算法 $doublepop(A, Top, i, y)$ 。



```
1  PROC Doublepush(VAR A:ARRAY(1..m);VAR Top:ARRAY(1..2);VAR i,x:Integer);
2  BEGIN
3      IF i≠1 AND i≠2 THEN 【
4          write("i error")
5      】
6      IF Top[1]=Top[2]-1 THEN 【
7          write("overflow") {栈满已不能入栈}
8      】 ELSE 【
```

```

9      Top[i]←Top[i]+(-1)^(i+1);
10     A[Top[i]]←x;
11   ]
12 END
13
14 PROC Doublepop(VAR A:ARRAY(1..m);VAR Top:ARRAY(1..2);VAR i,y:Integer);
15 BEGIN
16   IF i≠1 AND i≠2 THEN 【
17     write("i error")
18   】
19   IF (Top[i]=0 AND i=1) OR (TOP[i]=m+1 AND i=2) THEN 【
20     write("underflow"){栈空不能出栈}
21   】 ELSE 【
22     y←A[Top[i]];
23     Top[i]←Top[i]+(-1)^(i);
24   】
25 END

```

4. 对于采用A[m..n]组织的循环队列，试写一个统计队中数据元素个数的算法

```

1  FUNC GetCQLen(VAR A:ARRAY[m..n] OF datatype;front rear:integer):integer;
2  {统计并返回循环队列A[m..n]中的元素个数}
3  BEGIN
4    CASE
5      (front ≥d n) or (rear < m) or (front < m-1) or (rear > n):
6        【write('pointer linking error'); EXIT】;
7      front < rear: num←rear - front;
8      ELSE: front: rear→NIL;
9    ENDCASE
10   RETURN num;
11 END;

```

5.GDC Function

```

1  FUNC GCD(m,n:integer)
2  {辗转相除法求解两个非负整数m.n的最大公约数的算法}
3  BEGIN
4    IF n = 0 THEN 【
5      return m;
6    】 ELSE 【
7      VAR temp:m mod n
8      return GCD(n,temp);
9    】
10 END

```