数据结构第二节课官方笔记

目录

一、课件下载及重播方法

二、教材结构图

三、本章知识点及考频总结

（一）选择题（共 8 道）

1. 线性表（Linear List）是最简单和最常用的一种数据结构，它是由 **n** 个数据元素（结点）a_1, a_2, ...a_n 如组成的**有限序列**。其中，数据元素的个数 n 为表的长度。当 **n 为零时称为空表**，非空的线性表通常记为

（ a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n ）

这里的元素 $a_i(a\le i\le n)$是一个抽象的符号，它可以是一个数或者一个符号，还可以是较

复杂的记录。如一个学生、一本书等信息就是一个**数据元素**，它可以由若干个数据项组成。

2. 对于线性表，常见的基本运算有以下几种：

（1）置空表 InitList (L)，构造一个空的线性表 L。

（2）求表长 ListLength (L)，返回线性表 L 中元素个数，即表长。

（3）取表中第 i 个元素 GetNode (L, i)，若 1≤i≤ListLength (L)，则返回第 i 个元素 a_i。

（4）按值查找 LocateNode (L, x)，在表 L 中查找第一个值为 x 的元素，并返回该元素在表 L 中的位置，若表中没有元素的值为 x，则返回 0 值。

（5）插入 InsertList (L, i, X)，在表 L 的第 i 元素之前插入一个值为 x 的新元素，表 L 的长度加 1。

（6）删除 DeleteList (L, i)，删除表 L 的第 i 个元素，表 L 的长度减 1。

3. 一般来说，线性表的第 i 个元素 A 的存储位置为

$$LOC(a_i)=LOC(a_1)+(i-1)*d$$

其中，$LOC(a_1)$是线性表的第一个元素 a_1 的存储位置，通常称之为基地址。

线性表的这种机内表示称为线性表的顺序存储结构。它的特点是，元素在表中的相邻关系，在计算机内也存在着相邻的关系。

4. 线性表的插入运算是指在线性表的第 i-1 个元素和第 i 个元素之间插入一个新元素 x，使长度为 n 的线性表：

$$(a_1, a_2, \ldots, a_{i-1}, a_i, \ldots, a_n)$$

变为长度为 n+1 的线性表：

$$(a_1, a_2, \ldots, a_{i-1}, x, a_i, \ldots, a_n)$$

由于线性表逻辑上相邻的元素在物理结构上也是相邻的，因此在插入一个新元素之后，线性表的逻辑关系发生了变化，其物理存储关系也要发生相应的变化。除非 i=n+1，否则必须将原线性表的第 i、i+1、…、n 个元素分别向后移动 1 个位置，空出第 i 个位置以便插入新元素 x。

5. 一般情况下，在第 i（1≤i≤n）个元素之前插入一个新元素时，需要进行 n-i+1 次移动。而该算法的执行时间主要花在 for 循环的元素后移上，因此该算法的时间复杂度不仅依赖于表的长度 n，而且还与元素的插入位置 i 有关。当 i=n+1 时，for 循环一次也不执行，无需移动元素，属于最好情况，其时间复杂度为 O(1)；当 i=1，循环需要执行 n 次，即需要移动表中所有元素，属于最坏情况，算法时间复杂度为 O(n)。

6. 线性表的删除运算指的是将表中第 i 个元素删除，使长度为 n 的线性表

$$(a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n)$$

变为长度为 n-1 的线性表：

$$(a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$$

线性表的逻辑结构和存储结构都发生了相应的变化，与插入运算相反，插入是向后移动元素，而删除运算则是向前移动元素，除非 i=n 时直接删除终端元素，不需移动元素。

7. 该算法的时间复杂度分析与插入算法类似，删除一个元素也需要移动元素，移动的次数取决于表长 n 和位置 i。当 i=1 时，则前移 n-1 次；当 i=n 时不需要移动，因此算法的时间复杂度为 O(n)。由于算法中删除第 i 个元素是将从第 i+1 至第 n 个元素依次向前移动一个位置，共需要移动 n-i 个元素。

8. 线性表顺序存储结构的特点是，在逻辑关系上相邻的两个元素在物理位置上也是相邻的，因此可以随机存取表中任一元素。但是，当经常需要做插入和删除操作运算时，则需要移动大量的元素，而采用链式存储结构时就可以避免这些移动。然而，由于链式存储结

构存储线性表数据元素的存储空间可能是连续的，也可能是不连续的，因而链表的结点是不可以随机存取的。

（二）主观题（共 1 道）

1. 插入算法描述如下：

```
void InsertList ( SeqList *L, int i, DataType x )
{   //在顺序表 L 中第 i 个位置之前插入一个新元素 x
    int j;
    if ( i<1 || i>L->length+1 ) {
       printf ( "position error");
    return;
}
if ( L->length>=ListSize) {
    printf ( "overflow" );
    return;
}
for ( j=L->length-1;j>=i-1; j--)
    L->data[ j+1]=L->data[j];      //从最后一个元素开始逐一后移
L->data[ i-1]=x;                   //插入新元素 x
L->length++;                       //实际表长加 1
}
```

四、配套练习题

1、长度为 n 的顺序表，删除位置 i 上的元素(0≤i≤n-1)，需要移动的元素个数为（　　）

A: n-i

B: n-i-1

C: i

D: i+1

2、设顺序表首元素 A[0]的存储地址是 4000，每个数据元素占 5 个存储单元，则元素 A[20]的起始存储地址是（　　）

A: 4005

B: 4020

C: 4100

D: 4105

3、线性表是一个有限序列,组成线性表的基本单位是(　　)

A: 数据项

B: 数据元素

C: 数据域

D: 字符

[参考答案]：BCB

五、其余课程安排