

第一次 习题课

- 知识点小结
- 绪论
- 顺序表
- 链表

第一章涉及的知识点

- 什么是数据与数据结构
- 抽象数据类型及面向对象概念
 - ✿ 数据类型;
 - ✿ 数据抽象与抽象数据类型;
 - ✿ 面向对象的概念
 - ✿ 用于描述数据结构的语言

- 数据结构的抽象层次
- 算法的性能分析与度量
 - ◆ 算法的性能标准;
 - ◆ 算法的后期测试;
 - ◆ 算法的事前估计;
 - ◆ 空间复杂度度量;
 - ◆ 时间复杂度度量;

第二章： 需要复习的知识点

- 作为抽象数据类型的数组
 - ◆ 数组的定义和初始化;
 - ◆ 作为抽象数据类型的数组;
- 顺序表
 - ◆ 顺序表的定义和特点

- ◆ 顺序表的类定义
- ◆ 顺序表的查找、插入和删除算法
- ◆ 在顺序表中插入及删除时计算平均移动元素个数
- ◆ 使用顺序表的事例
- 稀疏矩阵
 - ◆ 稀疏矩阵的三元组表表示;
 - ◆ 稀疏矩阵的转置算法;

第三章 需要复习的知识点

■ 单链表

- ◆ 单链表的结构和类定义;
- ◆ 单链表中的插入与删除;
- ◆ 带表头结点的单链表;
- ◆ 用模板定义的单链表类;
- ◆ 静态链表

■ 单链表的算法

- ◆ 搜索含 x 结点

- ◆ 删除含 x 结点
- ◆ 统计单链表中结点个数
- 循环链表
 - ◆ 循环链表的类定义
 - ◆ 用循环链表解约瑟夫问题;
- 多项式及其相加
 - ◆ 多项式的类定义
 - ◆ 多项式的加法
- 双向链表
 - ◆ 双向循环链表的插入和删除算法

【例1】 设 n 是偶数， 且有如下程序段。

则语句 $y=y+i*j$ 的执行次数是多少？ 要求列出计算公式。

```
for(int i=1; i<=n; i++)  
    if(2*i<=n)  
        for(j=2*i; j<=n; j++)  
            y=y+i*j;
```


【解答】 执行次数是 $n^2/4$.语句

“ $y=y+i*j$ ”在 $i=1$ 时执行 $n-1$ 次，在 $i=2$ 时，执行 $n-3$ 次，以此类推，当 $i=n/2$ 时不再执行。故总的执行次数为 $(n-1)+(n-3)+\dots+3+1=n^2/4$

【例2】 已知长度为 n 的线性表 L 采用顺序存储结构，编写一个时间复杂度为 $O(n)$,空间复杂度为 $O(1)$ 的算法，该算法删除线性表中所有值为 x 的数据元素。

【解答】 执行用**K**记录顺序表**L**中不等于**x**的元素个数(即需要保存的元素个数), 边扫描边统计**k**, 并将不等于**x**的元素向前放在**k**位置上, 最后修改**L**的长度。对应的算法如下:

```
void delnode(sqList &L, ElemType x)
{   int k=0, i;
    for(int i=0; i<L.Length(); i++)
        if(L.data[i]!=x)
            {L.data[k]=L.data[i]; k++;}
}
```

【例3】 设稀疏矩阵M[6][7]如下图所示:(1)给出该图的三元组表A. (2)给出该图的转制矩阵的三元组表B. (3)当采用快速转置算法, 请写出辅助数组rowSize和rowStart

$$\begin{pmatrix} 0 & 0 & 0 & 9 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 2 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & -8 & 0 & 0 \\ 0 & 0 & 4 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

【解答】 (1)-(2)稀疏矩阵和它的转置的三元组显示在(a)和(b)中；

	行 (<i>row</i>)	列 (<i>col</i>)	值 (<i>value</i>)
[0]	0	3	9
[1]	1	1	3
[2]	2	2	7
[3]	2	3	2
[4]	3	0	5
[5]	3	4	-8
[6]	4	2	4
[7]	4	3	6
[8]	5	4	1

	行 (<i>row</i>)	列 (<i>col</i>)	值 (<i>value</i>)
[0]	0	3	5
[1]	1	1	3
[2]	2	2	7
[3]	2	4	4
[4]	3	0	9
[5]	3	2	2
[6]	3	4	6
[7]	4	3	-8
[8]	4	5	1

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	语 义
<i>rowSize</i>	1	1	2	3	2	0	0	矩阵 <i>A</i> 各列非 零元素个数
<i>rowStart</i>	0	1	2	4	7	9	9	矩阵 <i>B</i> 各行开 始存放位置

【例4】 有一个线性表(a_1, a_2, \dots, a_n), 采用带头节点的单链表L存储, 设计一个就地算法将其就地逆置。所谓“就地”是指算法的辅助空间应为 $O(1)$ 。

【解答】 对应的算法如下：

```
void Reversel(LinkList *&L)  
{   LinkList *p=L->link, *q;  
    L->link=NULL;  
    while(p!=NULL)  
    {q=p->link; p->link=L->link;  
      L->link=p; p=q;}  
}
```


【例5】 设有一个带头节点的双链表h，设计一个算法用于查找第一个元素值为x的节点，将其与后继结点进行交换。

【解答】 对应的算法如下：

```
void swap(DLinkedList *h, ElemType x)  
{   DLinkedList *p=h->rlink, *q;  
    while(p!=NULL && p->data!=x)  
        p=p->rlink;  
    if(p==NULL) return(0);  
    else { q=p->rlink;  
        if(q!=NULL)  
            {p->llink->rlink=q;  
                q->llink=p->llink;}  
    }
```

【解答】 对应的算法如下：

```
p->rlink=q->rlink; p->llink=q;  
if(q->rlink!=NULL) q->rlink->llink=p;  
q->rlink=p; return 1;}  
else return 0;}}
```

【例6】 设计一个算法用于带头节点的循环双链表是否对称。

【解答】 让p从左到右扫描，q从右向左扫描，直到它们指向同一节点或相邻为止，若它们所指节点值相同，则继续进行下去，否则返回0。若比较全相等，则返回1。对应的算法如下：

【解答】 对应的算法如下：

```
int symmetry(DLinkedList *h)  
{ DLinkedList *p=h->rlink, *q=h->llink;  
  while(p!=q && p->rlink!=q)  
    if(p->data==q->data)  
      {p=p->rlink; q=q->llink;}  
    else return 0;  
  return 1;  
}
```