



厦门大学《数据结构》期末试题·答案

考试日期：2011·1 (A)

信息学院自律督导部



一、(本题 10 分)

(1) 线性表和广义表的主要区别是什么?

(2) 已知广义表: $C=(a,(b,(a,b)),((a,b),(a,b)))$, 则 $\text{tail}(\text{head}(\text{tail}(C)))=?$

答案:

(1) 线性表和广义表都是元素 a_1, a_2, \dots, a_n 组成的序列, 其主要区别点在于: 在线性表中, a_i 是单个元素 (原子); 在广义表中, a_i 可以是单个元素 (原子), 也可以是广义表。(7 分)

(2) $\text{tail}(\text{head}(\text{tail}(C)))=((a,b))$ (3 分)

二、(本题 10 分) 简述二叉树的两种存储结构 (顺序存储和链式存储) 的数据结构及主要优缺点。在哈夫曼树中, 使用哪种存储结构, 并说明理由。

答案:

顺序存储结构: `typedef SqBiTree[Max_Tree_Size];`

特点: 使用数组存储二叉树上的结点元素, 按照对应的完全二叉树的编号来存储二叉树。优点是适用于完全二叉树, 访问方便。缺点是对一般二叉树, 较大地浪费了空间。(4 分)

链式存储结构:

`typedef struct BiTNode{ TElemType data; struct BiTNode *lchild, *rchild;} BiTNode, *BiTree;`

特点: 使用结构体来表示结点元素, 使用指针来指向结点的左右孩子。优点是插入与删除方便, 节省空间, 缺点是不能快速地随机访问结点元素。(4 分)

在哈夫曼树中, 使用静态三叉链表, 这样可以方便地从根走到叶子, 也可以从叶子走到根, 而且可以随机访问和节省空间。(2 分)

三、(本题 10 分) 一棵二叉树的先序、中序和后序序列分别如下, 其中有一部分未显示出来, 试求出空格处的内容, 并画出该二叉树。

先序序列: B F I C E H G ; 中序序列: D K F I A H E J C G ; 后序序列: K F B H J E G C A 。

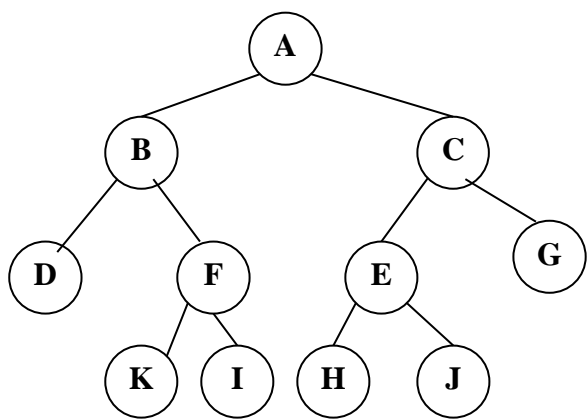
答案:

先序序列: A B D F K I C E H J G

中序序列: D B K F I A H E J C G

后序序列: D K I F B H J E G C A (11 分)

画出树得 4 分。



四、（本题 10 分） 分别使用普里姆算法和克鲁斯卡尔算法求出图 G1 的最小生成树， 仅需画出最小生成树的成长过程即可。

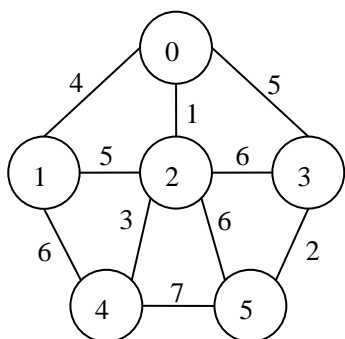
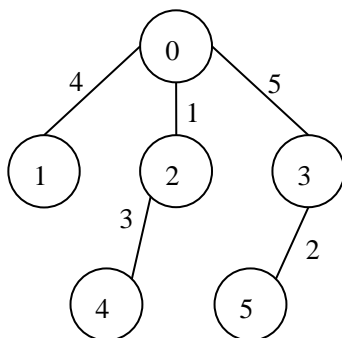
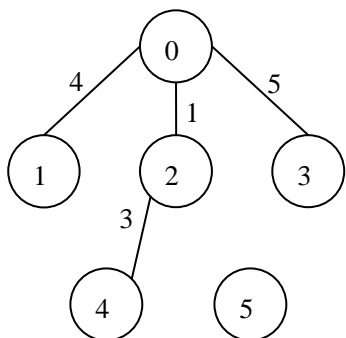
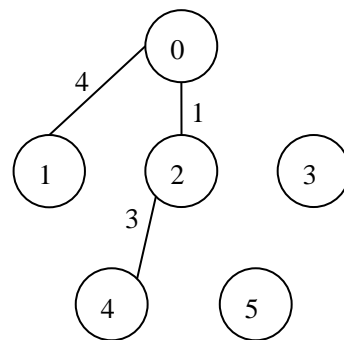
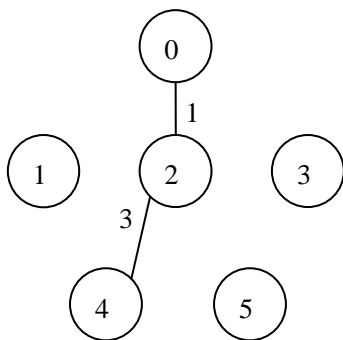
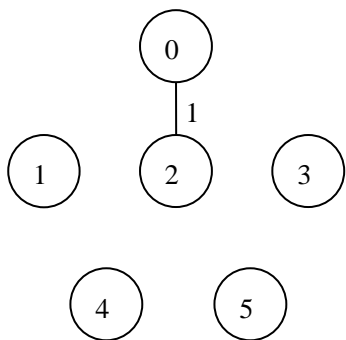


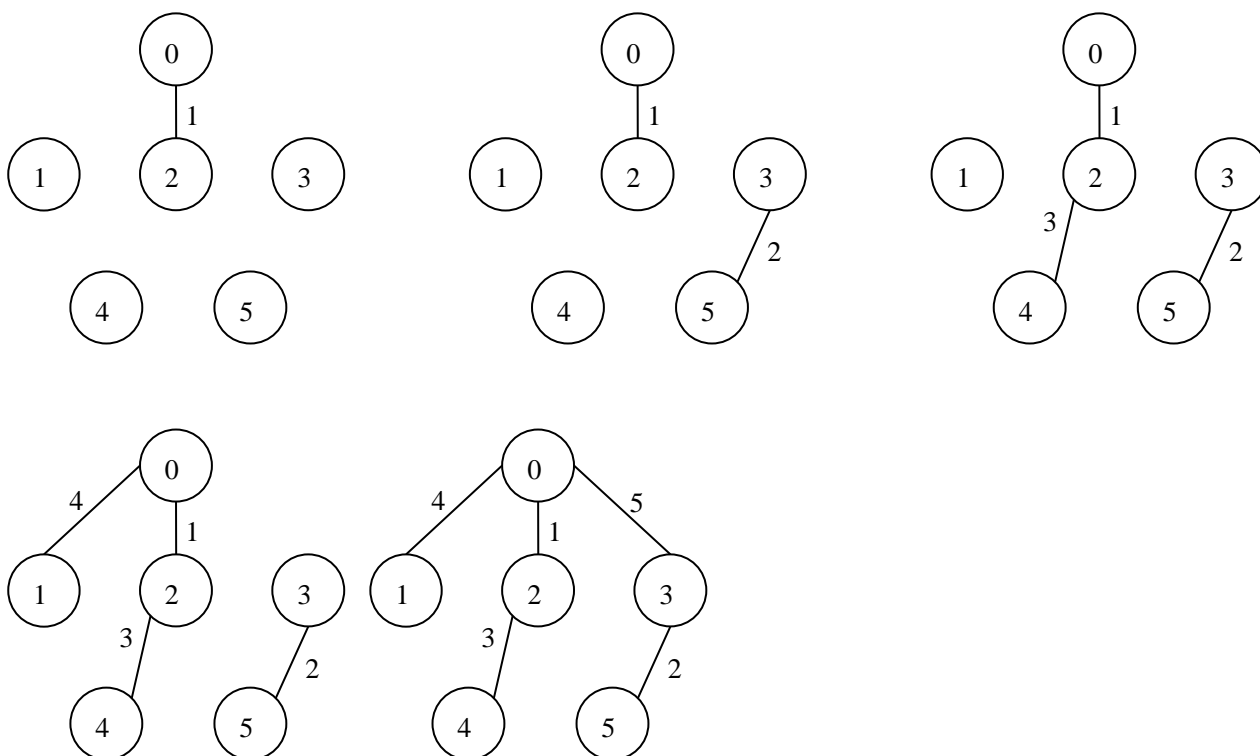
图 G1

答案：

（1）普里姆算法求最小生成树的过程如下： 5 分

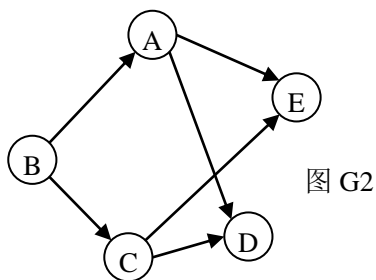


(2) 克鲁斯卡尔算法如下：5 分



五、(本题 10 分) 有向图 G2 如上所示，

- (1) 请写出图 G2 所有可能的拓扑序列：
- (2) 请写出以顶点 B 为起始点的深度优先遍历序列和广度优先遍历序列，并画出对应的生成树。遍历过程中当有多种选择时，编号小的结点优先。



答案：

(1) BACDE、BACED、BCADE、BCAED (5 分，少一个扣一分)

(2) 深度优先遍历序列：BADEC (3 分)

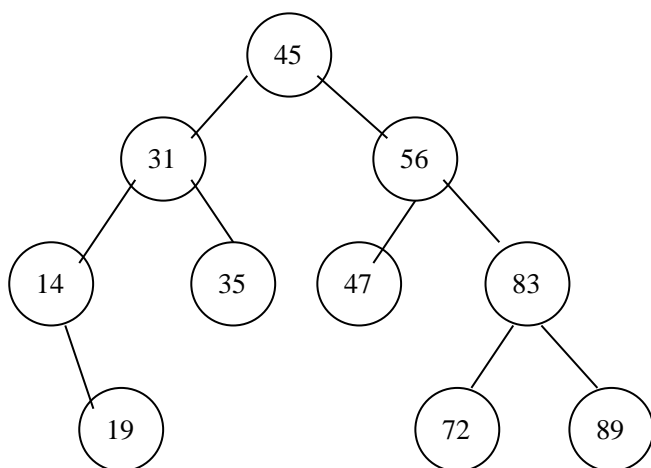
广度优先遍历序列：BACDE (3 分)

六、(本题 15 分) 已知键值序列为 {45,56,83,31,72,35,14,47,89,19}，要求给出：

- (1) 按键值排列次序构造一棵二叉排序树。
- (2) 在等概率的情况下，求出该二叉排序树查找成功的平均查找长度。
- (3) 针对上述 10 个键值，在不同的排列次序下所构造出的不同形态的二叉排序树中，在最坏和最好情况下，二叉排序树的高度各是多少？

答案：

(1)



(2) 在等概率情况下，该二叉排序树的平均检索长度是：

$$ASL = (1 + 2 \times 2 + 3 \times 4 + 4 \times 3) / 10 = 29 / 10 = 2.9$$

(3) 对于上述 10 个键值，在最坏情况下，每个结点（除了叶子结点）只有右孩子（或者只有左孩子），高度为 10。在最好情况下，高度为 $\lfloor \log_2 10 \rfloor + 1 = 4$ 。

七、（本题 15 分）设关键字序列为：49,38,66,80,70,15,22，欲对该序列进行从小到大排序。

(1) 用直接插入排序法进行排序，写出每趟的结果。

(2) 采用待排序列的第一个关键字作为枢轴，写出快速排序法的一趟和二趟排序之后的状态。

(3) 画出待排序列的初始化堆。

答案：

① 直接插入排序法

原始关键字序列为：(49) 38 66 80 70 15 22

(38 49) 66 80 70 15 22

(38 49 66) 80 70 15 22

(38 49 66 80) 70 15 22

(38 49 66 70 80) 15 22

(38 49 66 70 80) 15 22

(15 38 49 66 70 80) 22

(15 22 38 49 66 70 80)

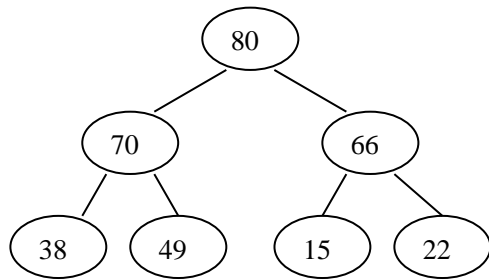
② 快速排序法

原始关键字序列为：49, 38, 66, 80, 70, 15, 22

第一趟排序后 22 38 15 (49) 70 80 66

第二趟排序后 15 (22) 38 66 (70) 80

③ 该堆是最大堆，具体如下：



八、（本题 10 分）假设一棵树的存储结构采用双亲表示法，双亲数组为 `int parent[MaxSize]`，其中 `MaxSize` 为最大结点个数。树中各结点按先根遍历的次序存放，根结点存于 `parent[0]`。试编写一个函数，计算下标 `p` 所指结点和下标 `q` 所指结点的最近公共祖先结点。

参考答案：

```

int CommonAncestry(int parent[], int MaxSize, int p, int q){
    int i,j;
    for (i=p; i!=-1; i=parent[i])
        for (j=q; j!=-1; j=parent[j])
            if (i==j) return i;
}

```

九、（本题 10 分）1, 2, …, n 这 n 个数，无序地保存在数组 `c[1..n]` 中。请编写一个时间复杂度为 $O(n)$ 的排序算法，将数组 `c[1..n]` 按小到大排序。

参考答案：

```

void C_sort(int c[], int n)
{
    int i, x;
    for (i=1; i<=n; i++)
        while (c[i]!=i){
            x=c[i];
            c[i]=c[x];
            c[x]=x;
        }
}

```

交换 $O(n)$ 次。