



厦门大学《数据结构》期末试题·答案

考试日期：2010.1 (zch)

信息学院自律督导部



一、(本题 10 分)

(1) 简述线性表的两种存储结构的主要优缺点及各自适用的场合。

(2) 在折半查找和表插入排序中，记录分别应使用哪种存储结构，并用一句话简述理由。

答：

(1) 顺序存储是按索引（如数组下标）来存取数据元素，优点是可以实现快速的随机存取，缺点是插入与删除操作将引起元素移动，降低效率。对于链式存储，元素存储采取动态分配，利用率高。缺点是须增设指针域，存储数据元素不如顺序存储方便。优点是插入与删除操作简单，只须修改指针域。

(2) 在折半查找中，记录使用顺序存储，可以快速实现中点的定位；在表插入排序中，记录使用静态链表，可以减少移动记录的操作。

二、(本题 15 分) 在带头结点的非空线性链表中，试设计一算法，将链表中数据域值最小的那个结点移到链表的最前面，其余各结点的顺序保持不变。要求：不得额外申请新的链结点。

解：程序如下：

```
typedef struct node {
    int data;
    struct node * next;
}Node,*LinkList;
void MinFirst(LinkList L)
{Node *p,*q,*ptrmin;
    if(L->next == NULL) return; //空表
    ptrmin = L; //ptrmin 指向当前最小结点的前一个结点
    p = L->next; //p 指向当前结点的前一个结点
    while(p->next!=NULL) {
        if( p->next->data < ptrmin->next->data ) ptrmin = p;
        p = p->next;
    }
    //q 指向最小结点，并从链表中删除
    q = ptrmin->next; ptrmin->next = q->next;
    q->next = L->next; L->next = q; //q 指向的最小结点插入到链表头
```

三、(本题 10 分) 证明：一棵二叉树的先序序列和中序序列可惟一确定这棵二叉树。

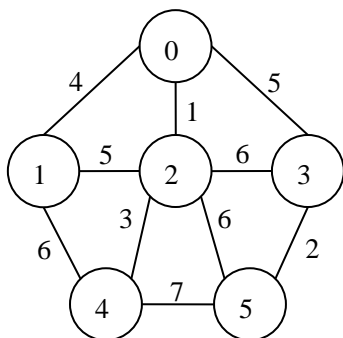
证明：A 类题，15 分。

设一棵二叉树的先序序列和中序序列分别存放在一维数组 $A[1..n]$ 和 $B[1..n]$ 中。因为先序序列的第一个结点 $A[1]$ 为二叉树的根结点，在中序序列中找到与 $A[1]$ 相同的结点，不妨假设

$B[i]=A[1]$ ；又因为二叉树的任何一棵子树的结点是紧挨在一起的，故所构造的二叉树的左子树由先序序列 $A[2..i]$ 和中序序列 $B[1..i-1]$ 确定的二叉树组成，而所构造的二叉树的右子树由先序序列 $A[i+1..n]$ 和中序序列 $B[i+1..n]$ 确定的二叉树组成。这是一个递归过程，当先序序列和中序序列分别含有 3 个以下的结点时，可惟一确定对应的二叉树。因此，由一棵二叉树的先序序列和中序序列可以惟一确定这棵树。

四、（本题 15 分）给定下面的带权无向图 G ：

- 1) 从顶点 0 开始，请写出深度优先遍历序列和广度优先遍历序列，当有多种选择时，编号小的结点优先。
- 2) 从顶点 0 开始，使用普里姆算法求出该图的最小生成树，需画出最小生成树的构造过程。
- 3) 有人给出求解最小生成树的另外一种算法：将连通图中的边按其权值从大到小顺序逐个删除直至不可再删，删除要遵循的原则是：保证在删除该边后各个顶点之间应该是连通的。请问该算法是正确的吗？如果认为是正确的，请给出证明。如果是错误的，请给出反例。

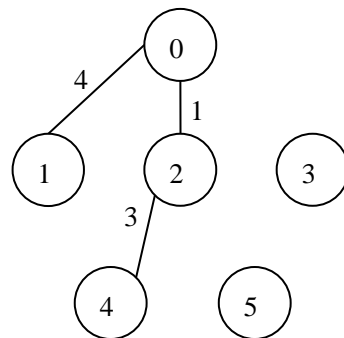
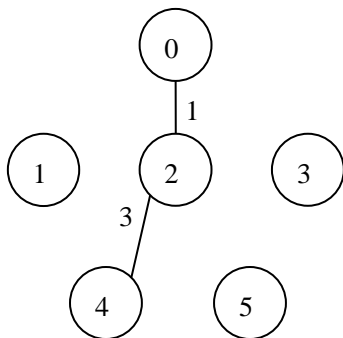
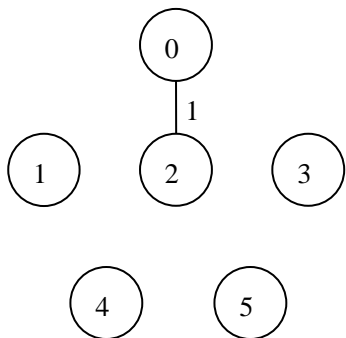


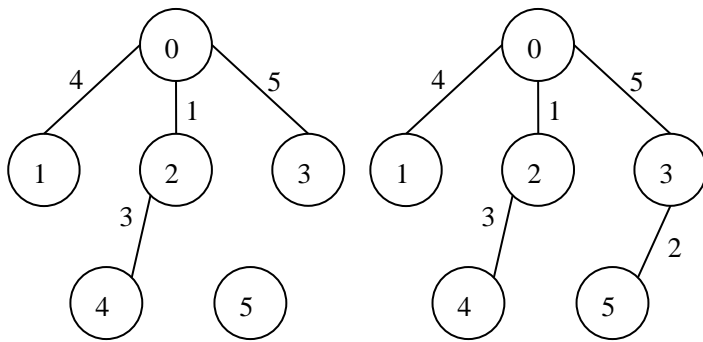
解：

(1) 深度优先遍历序列为：0, 1, 2, 3, 5, 4

广度优先遍历序列为：0, 1, 2, 3, 4, 5

(2) 普里姆算法求最小生成树的过程如下：





(3) 该算法是正确的，称为破圈法。只要证明：图中若含有回路，则回路中权值最大的边一定不在最小生成树上。可以使用反证法。

五、(本题 10 分) 已知待散列存储的关键字序列为 (4,15,38,49,33,60,27,71)，哈希函数为 $H(\text{key}) = \text{key} \bmod 11$ ，哈希表 HT 的长度为 11，采用线性探测再散列法解决冲突。试构造此哈希表，并求出在等概率情况下查找成功的平均查找长度。

解：哈希表为

0	1	2	3	4	5	6	7	8	9	10
33				4	15	38	49	60	27	71

平均查找长度为： $(1+2+2+3+1+4+5+6)/8 = 3$ 。

六、(本题 15 分) 以关键字序列 (29, 18, 25, 47, 58, 12, 51, 10) 为例，执行以下排序算法，写出每一趟结束时的关键字状态：

(1) 增量序列为 5, 3, 1 的希尔排序 (2) 快速排序 (3) 堆排序。

解：(1)

第 1 趟：12, 18, 10, 29, 47, 58, 51, 25

第 2 趟：12, 18, 10, 29, 25, 58, 51, 47

第 3 趟：10, 12, 18, 25, 29, 47, 58, 51

(2)

第 1 趟：10, 18, 25, 12, 29, 58, 51, 47

第 2 趟：10, 18, 25, 12, 29, 47, 51, 58

第 3 趟：10, 12, 18, 25, 29, 47, 51, 58

第 4 趟：10, 12, 18, 25, 29, 47, 51, 58

(3)

第 1 趟：58, 47, 51, 29, 18, 12, 25, 10

第 2 趟：51, 47, 25, 29, 18, 12, 10, (58)

第 3 趟：47, 29, 25, 10, 18, 12, (51, 58)

第 4 趟：29, 18, 25, 10, 12, (47, 51, 58)

第 5 趟：25, 18, 12, 10, (29, 47, 51, 58)

第 6 趟：18, 10, 12, (25, 29, 47, 51, 58)

第 7 趟: 12, 10, (18, 25, 29, 47, 51, 58)

第 8 趟: 10, (12, 18, 25, 29, 47, 51, 58)

七、(本题 10 分) 在两个有序线性表中, 寻找是否存在共同元素。如果存在共同元素, 返回第一个共同元素在第一个有序表中的位置。请设计数据结构, 并在其上设计算法。

答: 可以参考有序表的归并算法。

数据结构可以使用一维数组, 并且第 0 个元素放空。

int SearchCommonItem(int a[n], int b[m])//第 0 位放空, 返回值为 0 代表找不到

```
{
    int i=1,j=1;
    while (i<=n && j<=m)
    {
        if (a[i]==b[j]) return i;
        else if (a[i]<b[j]) i++;
        else j++;
    }
    return 0;
}
```

八、(本题 15 分) 给出一系列整数, 设计算法求出总和最大的子序列, 要求算法的时间复杂性在 $O(n)$ 之内。

答:

int maxsubsum(int a[n])

```
{
    int maxsum=0; thissum=0; int j=0;
    for (j=0; j<n; j++)
    {
        thissum=thissum+a[j];
        if (thissum>maxsum) maxsum=thissum;
        else if (thissum<0) thissum=0;
    }
    return maxsum;
}
```

时间复杂性为 $O(n)$ 。