



厦门大学《数据结构》期末试题·答案

考试日期：2010·1 (A)

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。

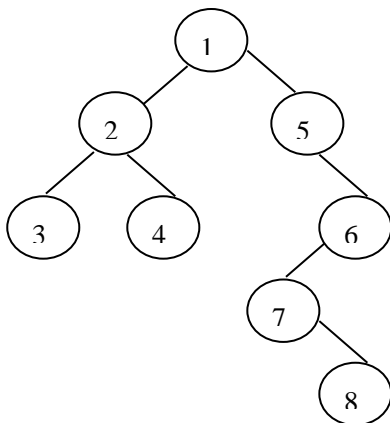
(2) 在折半查找和表插入排序中，记录分别应使用哪种存储结构，并用一句话简述理由。

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 10 分) 一棵二叉树的先序、中序和后序序列分别如下，部分未显示，请画出该二叉树。先序序列：_2 3_5_7 8; 中序序列：3_4 1_7 8 6; 后序序列：_4 2 __6 5 1。

答：



三、(本题 10 分) 假定用于通讯的电文仅由 a、b、c、d、e、f、g 等 8 个字母组成，字母在电文中出现的频率分别为：0.07、0.19、0.02、0.06、0.32、0.03、0.21 和 0.10。试为这些字母设计哈夫曼编码。

答：一种编码如下：

a: 0010 b: 10 c: 00000 d: 0001 e: 01 f: 00001 g: 11 h: 0011

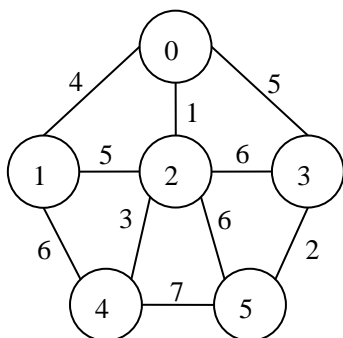
四、(本题 15 分) 给定下面的带权无向图 G：

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。

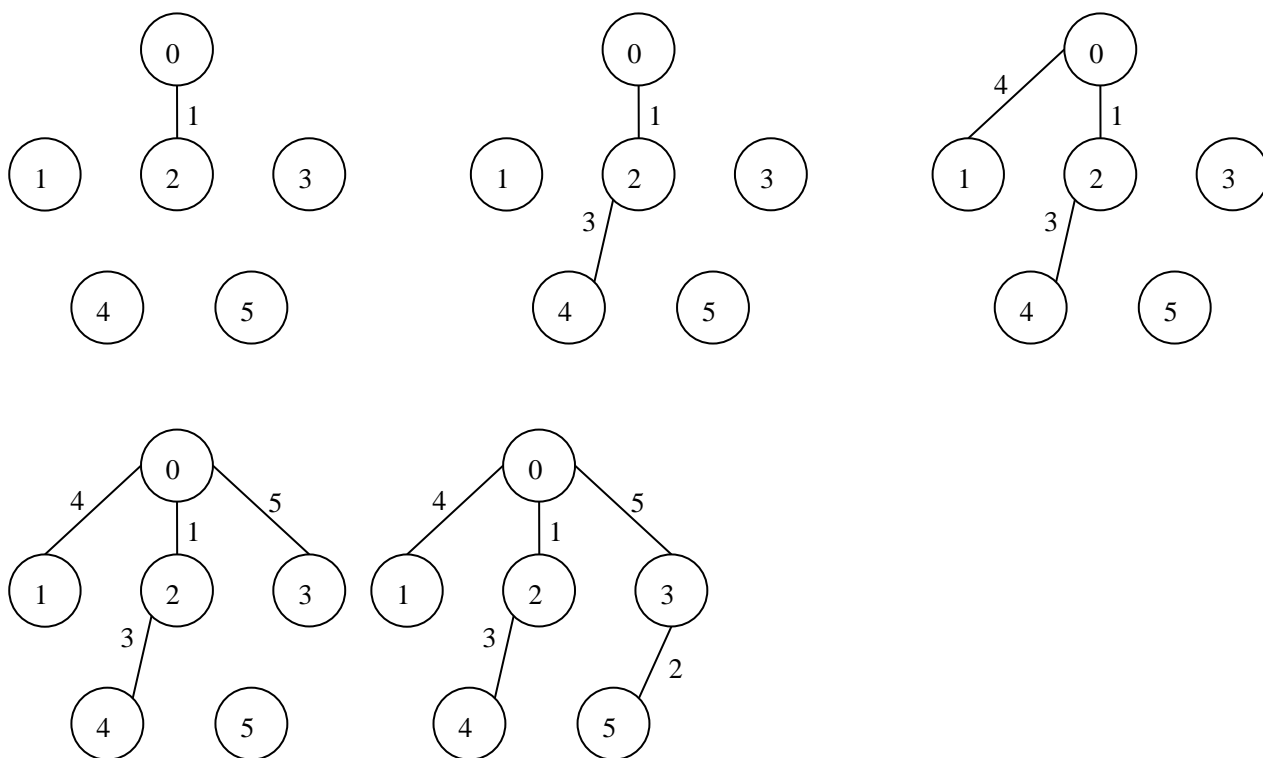
3) 有人给出求解最小生成树的另外一种算法：将连通图中的边按其权值从大到小顺序逐个删

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。



解答：

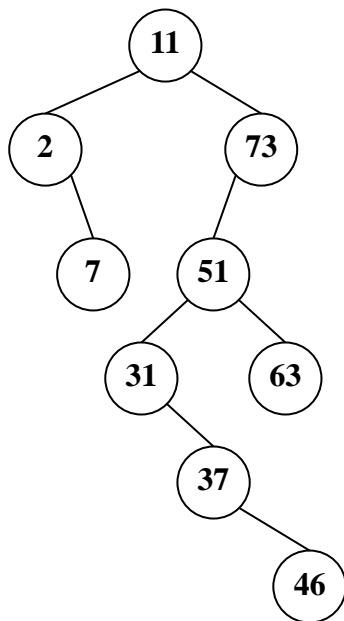
- (1) 深度优先遍历序列为：0, 1, 2, 3, 5, 4
 广度优先遍历序列为：0, 1, 2, 3, 4, 5
 (2) 普里姆算法求最小生成树的过程如下：



(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 15 分) 设有一个关键字序列{11,73,51,31,63,37,46,2,7}，(1) 从空树开始构造排序二叉树，画出得到的排序二叉树；(2) 计算该排序二叉树在等概率下查找成功的平均查找长度。(3) 如果要按照从大到小的顺序访问关键字，请简要说明遍历方法。

答：(1)



(2) $(1+4+6+8+5+6)/9=10/3$

(3) 对该排序二叉树进行：先遍历右子树，再遍历根，后遍历左子树。

六、(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，写出用快速排序法的一趟和二趟排序之后的状态。

(3) 假设有个系统要多次对 n 个关键字进行排序， n 很大且每次排序时关键字的分布情况不明。系统不希望每次排序时间变动过大，而且希望越快越好，哪种排序算法较好？为什么？

答：

(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) 堆排序算法较好。首先，因为堆排序平均时间复杂度为 $O(n\log n)$ ，性能较好；另外，因为关键字分布情况不明，堆排序最坏时间复杂度为 $O(n\log n)$ ，保证计算时间不会出现 $O(n^2)$ 而造成很大的延迟，此时，快速排序算法不太合适。此外，堆排序只需要 $O(1)$ 个辅助空间，归并排序虽然总体性能也很不错，也满足系统的要求，但需要 $O(n)$ 的辅助空间。

七、（本题 10 分）设 L 是一个带头结点的非递减有序单链表的表头指针，试设计一个算法，将元素 e 插入到链表 L 中的合适地方，使得该链表仍是非递减有序。

答：

```
typedef struct Node {
    ElemType e;
    struct Node * next;
} Node, *LinkList;
void Insert( LinkList * L, ElemType e)
{ Node * p, *q, *s;
  p = L->next; q = L;
  while ( p!=NULL && p->data <= e )
    q = p; p = p->next;
  }
  s = (Node *) malloc( sizeof(Node) );
  s->data = e;
  s->next = p;
  q->next = s;
}
```

八、（本题 15 分）给出一系列整数，设计算法求出总和最大的子系列，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 $O(n)$ 。