中国地质大学（武汉）考试出题专用纸

教务处制

试卷类别

A

使用班级

191081-4

192081-3

193081-2

使用学期

2010 春

任课教师

系主任

审核签字

考试课程名称：数 据 结 构　　学时：56

考试方式：闭卷，笔试

考试内容：

一、单项选择题（每题 2 分，共 40 分）

1．以下哪一个术语与数据的存储结构无关？（　　）

A．栈　　B．哈希表　　C．线索树　　D．双向链表

2．以下算法的时间复杂度是（　　）。

```
int func(int n)
 { int x=1;
   while (x*x<=n) x++;
   return x;
  }
```

A．$O(n^2)$　　B．$O(\sqrt{n})$　　C．$O(\log_2 n)$　　D．$O(2^n)$

3．若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用（　　）存储方式最节省时间。

A．顺序表　　B．双链表　　C．带头结点的双循环链表　　D．单循环链表

4．链表不具有的特点是（　　）。

A．插入、删除不需要移动元素　　B．可随机访问任一元素

C．不必事先估计存储空间　　D．所需空间与线性长度成正比

5．在单链表指针为 p 的结点之后插入指针为 s 的结点，正确的操作是（　　）。

A．p->next=s; s->next=p->next;　　B．s->next=p->next; p->next=s;

C．p->next=s; p->next=s->next;　　D．p->next=s->next; p->next=s;

6．若一个栈的输入序列为 1,2,3,…,n，输出序列的第一个元素是 i，则第 j 个输出元素是（　　）。

A. i-j-1　　B. i-j　　C. j-i+1　　D. 不确定的

7．数组 A[0..5,0..6]的每个元素占五个字节，将其按列优先次序存储在起始地址为 1000 的内存单元中，则元素 A[5,5]的地址是(　　)。

A. 1175　　B. 1180　　C. 1205　　D. 1210

8．在下述结论中，正确的是（　　）。

①只有一个结点的二叉树的度为 0；

②二叉树的度为 2；

③二叉树的左右子树可任意交换；

④深度为 K 的完全二叉树的结点个数小于或等于深度相同的满二叉树。

A．①②③　　B．②③④　　C．②④　　D．①④

9．若一棵二叉树具有 10 个度为 2 的结点，5 个度为 1 的结点，则度为 0 的结点个数是（　　）。

A．9　　B．11　　C．15　　D．不确定

10．设森林 F 中有三棵树，第一，第二，第三棵树的结点个数分别为 M1，M2 和 M3。与森林 F 对应的二叉树根结点的右子树上的结点个数是（　　）。

A．M1　　B．M1+M2　　C．M3　　D．M2+M3

11．n 个结点的线索二叉树上含有的线索数为（　　）。

A．2n　　B．n－1　　C．n＋1　　D．n

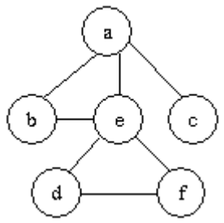
12．下述二叉树中,哪一种满足性质:从任一结点出发到根的路径上所经过的结点序列按其关键字有序（　）。

A．二叉排序树　B．哈夫曼树　　C．平衡二叉树　　D．堆

13．设图如下所示，在下面的 5 个序列中，符合深度优先遍历的序列有多少？（　　）

a e b d f c　　a c f d e b　　a e d f c b　　a e f d c b　　a e f d b c

A．5 个　　B．4 个　　C．3 个　　D．2 个



14．下面关于求关键路径的说法不正确的是（　　）。

A．求关键路径是以拓扑排序为基础的

B．一个事件的最早开始时间同以该事件为尾的弧的活动最早开始时间相同

C．一个事件的最迟开始时间为以该事件为尾的弧的活动最迟开始时间与该活动的持续时间的差

D．关键活动一定位于关键路径上

15．在有向图 G 的拓扑序列中，若顶点 Vi 在顶点 Vj 之前，则下列情形不可能出现的是(　　)。

A．G 中有弧<Vi,Vj>　　B．G 中有一条从 Vi 到 Vj 的路径

C．G 中没有弧<Vi,Vj>　　D．G 中有一条从 Vj 到 Vi 的路径

16．具有 12 个关键字的有序表，折半查找的平均查找长度（　　）。

A. 3.1　　B. 4　　C. 2.5　　D. 5

17．分别以下列序列构造二叉排序树，与用其它三个序列所构造的结果不同的是(　　)。

A．（100, 80, 90, 60, 120, 110, 130）　B．（100, 120, 110, 130, 80, 60, 90）

C．（100, 60, 80, 90, 120, 110, 130）　D．(100, 80, 60, 90, 120, 130, 110)

18．　m 阶 B-树是一棵(　　) 。

A. m 叉排序树　B. m 叉平衡排序树　C.　m-1 叉平衡排序树　D.　m+1 叉平衡排序树

19．下列排序算法中，其中（　　）是稳定的。

A. 堆排序，冒泡排序　　B. 快速排序，堆排序

C. 直接选择排序，归并排序　　D. 归并排序，冒泡排序

20．下列排序算法中，时间复杂度为 $O(n\log_2 n)$且占额外空间最少的是（　　）。

A．堆排序　　B.冒泡排序　　C.快速排序　　D.希尔排序

中国地质大学（武汉）考试出题专用纸

教务处制

试卷类别

A

使用班级

191081-4

192081-3

193081-2

使用学期

2010 春

任课教师

蔡之华

朱晓莲

李桂玲

系主任

审核签字

二、解答题（共 35 分）

1.（8 分）证明：由一棵二叉树的前序序列和中序序列可唯一确定这棵二叉树。设一棵二叉树的前序序列为 ABDGECFH，中序序列为 DGBEAFHC，试画出该二叉树。

2.（9 分）采用哈希函数 H(k)=(3*k) mod 13 并用线性探测开放地址法处理冲突，在数列地址空间［0..12］中对关键字序列 22,41,53,46,30,13,1,67,51

（1）构造哈希表，画出示意图；

（2）装填因子；

（3）计算等概率下查找成功的平均查找长度。

3.（10 分）已知待排序的序列为（503，87，512，61，908，170，897，275，653，462）。

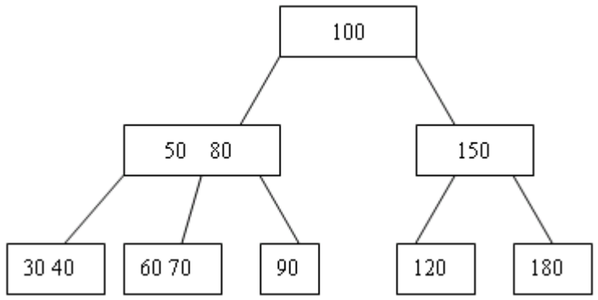
（1）根据以上序列建立一个堆，希望先输出最小值；

（2）输出最小值后，如何得到次小值，并画出相应结果图。

4.（8 分）设有 3 阶 B-树如下图所示。

（1）画出在原 B-树中插入关键字 20 后的 B-树；

（2）画出在原 B-树中删除关键字 150 后的 B-树。



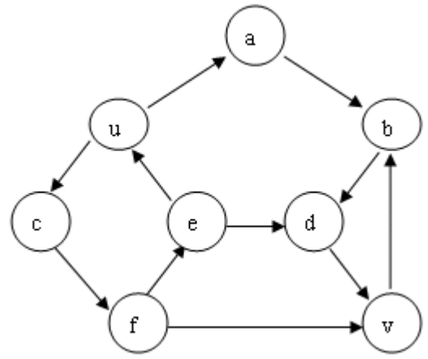
三、算法设计题（共 25 分）

1.（10 分）已知不带头结点的线性链表 list，链表中结点构造为（data, link），其中 data 为数据域，link 为指针域。请写一算法，将该链表按结点数据域的值的大小从小到大重新链接。要求链接过程中不得使用除该链表以外的任何链结点空间。

2.（15 分）假设图 G（如下图所示）采用邻接表存储，设计一个算法，输出图 G 中从顶点 u 到顶点 v 长度为 k 的所有简单路径。

（1）说明算法的基本思想；

（2）编写算法，允许用 C/C++/Java 来描述，在算法关键的地方给出必要的注释。



2010 上《数据结构》A 卷 参考答案

（191081-4、192081-3、193081-2 班用）

一、选择题（每题 2 分）

ABABB　DADBD　CDDCD　ACBDA

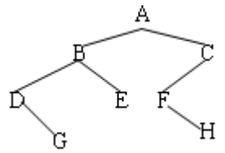
二、解答题（共 35 分）

1.（8 分）

（1）证明：

给定二叉树结点的前序序列和对称序（中序）序列，可以唯一确定该二叉树。因为前序序列的第一个元素是根结点，该元素将二叉树中序序列分成两部分，左边（设 l 个元素）表示左子树，若左边无元素，则说明左子树为空；右边（设 r 个元素）是右子树，若为空，则右子树为空。根据前序遍历中“根—左子树—右子树”的顺序，则由从第二元素开始的 l 个结点序列和中序序列根左边的 l 个结点序列构造左子树，由前序序列最后 r 个元素序列与中序序列根右边的 r 个元素序列构造右子树。

（2）二叉树为：



2.（9 分）

（1）

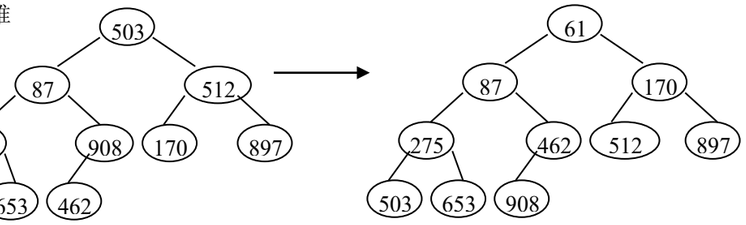
散列地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键字	13	22		53	1		41	67	46		51		30
比较次数	1	1		1	2		1	2	1		1		1

（2）装填因子=9/13=0.69

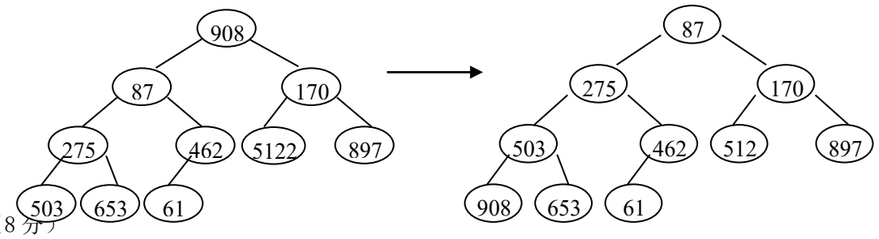
（3）ASL_{succ} =11/9

3.（10 分）

(1)建小堆

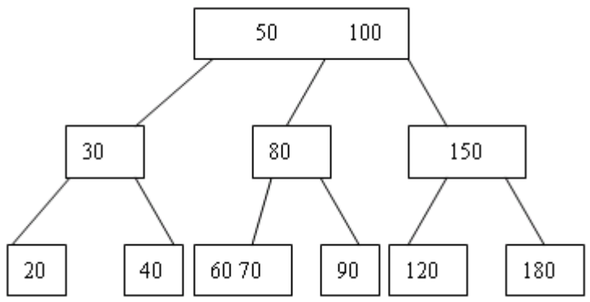


(2) 求次小值

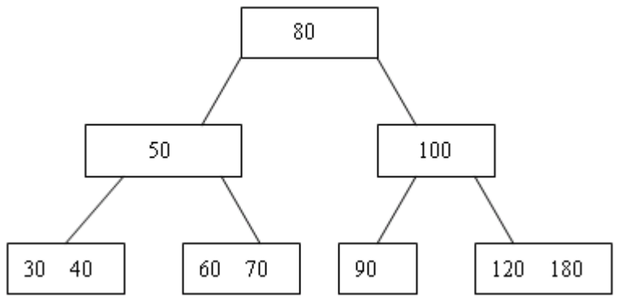


4.（8 分）

（1）插入关键字 20 后的 B-树



（2）删除关键字 150 后的 B-树



三、算法设计题（共 25 分）

1. (10 分)

```
typedef struct Node
   {   DataType data;
         struct Node *link;
      }SLNode;
SLNode * LinkListSort(SLNode * list)
{
         SLNode * p, *q, *r;
         p=list->link;     //p 是工作指针，指向待排序的当前元素。
         list->link=NULL;//假定第一个元素有序，即链表中现只有一个结点。
         while(p!=NULL)
          {  r=p->link;       //r 是 p 的后继。
             q=list;
         if(q->data>p->data)//处理待排序结点 p 比第一个元素结点小的情况。
         {   p->link=list;
            list=p;//链表指针指向最小元素。
          }
         else//查找元素值最小的结点。
          {   while(q->link!=NULL && q->link->data<p->data) q=q->link;
             p->link=q->link;//将当前排序结点链入有序链表中。
             q->link=p;     }
       p=r; //p 指向下个待排序结点。
      }
   }
```

2.（1）（4 分）

算法思想：采用带回溯的深度优先搜索算法。

用全局变量 path[Max]记录到达当前结点时所走过的路径，作用是输出该路径；用另一个全局变量记录当前走过的路径长度。初始 path 变量为空，每向下走一步都把对应的结点编号放到 path 变量中并标记为已访问状态，以避免回路。

（2）（11 分）

```
#define Max 20
typedef struct arc
   { int adjvex;       /*邻接点在数组中的序号*/
        struct arc *nextarc; /*链域，指示下一条边或弧*/
    }arcnode;
typedef struct
  {   DataType data;                  /*顶点信息*/
       arcnode *firstarc;         /*指示第一个邻接点*/
   }vnode;
```

```
int path[Max];          //path 存放当前走过的路径上的结点
int visited[Max];      //visited 标记各结点是否被访问过
int i;
for(i=0; i<k; i++)
          visited[i]=0;
void getAllPath(vnode G[ ],int vi, int vj, int k, int len)
{
   arcnode * p;
   int w;
   if (len>=k) return; //如果当前路径长度大于 k，则直接退出
   visited[vi]=1;       //标记当前的结点已被访问
   len++;                  //当前路径长度加 1
   path[len]=vi;        //把当前结点 vi 放到路径数组中
   if (vi= =vj && len= =k)   //找到了一条路径且长度符合要求
         printPath();     //打印该路径，或保存起来
   else if (len>0)        //若没退回到初始结点，则继续
   {
       p=algraph[vi].firstarc;
       while (p!=NULL)
         {
            w=p->adjvex;
            if (!visited[w])
                  getAllPath(G,w,vj,k,len);    //递归调用
            p=p->nextarc;
         }
   }
   visited[vi]=0;   //取消访问标志，使该结点可以重用
   len- -;
}
//其他函数调用函数 getAllPath();
getAllPath(G, u, v, k,-1);
```