考试科目：数 据 结 构 （共 150 分）

一、选择题：（每小题 3 分，共 15 分）

在下列备选答案中选出一个正确的，将其号码填在“_____”上。

1. 分别以下列序列构造二叉排序树，与众不同的是_____。

a. 100,80,60,85,110,120,150　　b. 100,80,60,85,120,110,150

c. 100,80,85,60,120,110,150　　d. 100,80,60,85,120,150,110

2. 对数据序列（8,9,10,4,5,6,20,1,2）采用（由后向前次序的）冒泡排序，需要进行的趟数（遍数）至少是_____。

a. 3　　b. 4　　c. 5　　d. 8

3. 在图采用邻接表存储时，求最小生成树的 Prim 算法的时间复杂度为_____。

a. O(n)　　b. O(n + e)　　c. $O(n^2)$　　d. $O(n^3)$

4. 已知一棵二叉树的先序序列和中序序列相反，则该二叉树一定满足_____。

a. 左子树为空　　b. 其中任一结点无左孩子

c. 右子树为空　　d. 其中任一结点无右孩子

5. 一棵左右子树均不空的二叉树在先序线索化后，其中值为空的链域的个数是_____。

a. 0　　b. 1　　c. 2　　d. 不确定

二、判断题：（每小题 1.5 分，共 15 分）

判断下列各题是否正确，若正确，在（）内打“√”，否则打“×”。

1.（ ）链表中元素之间的逻辑次序是由其指针确定的。

2.（ ）在链队列中执行出队操作是在队头进行的，故不可能改变尾指针的值。

3.（ ）在顺序栈结构中，如果将栈顶放在数组的开头的位置不会影响运算的时间性能。

4.（ ）在二叉树顺序存储结构中（根的下标为 1），下标为 130 的结点一定处于左子树中。

5.（ ）在一棵左右非空的二叉树中，根结点的中序前趋不一定是叶子结点。

6.（ ）如果无向图的深度遍历序列唯一，则可唯一确定出该图。

7.（ ）一个图的最小生成树可能不唯一，但权值最小的边一定会出现在所有的解中。

8.（ ）如果散列表中关键字不同的两个元素的散列函数相同，则称这两个元素为同义词。

9.（ ）在 B-树中，任意分支结点中的关键字的前趋一定在叶子结点中。

10.（ ）快速排序算法在初始数据表为有序时的时间性能达到最好。

三、填空（每空 3 分,共 30 分）：

1. 判断带头结点的单循环链表 L 仅有一个元素结点的条件是

_____ _____ 。

2. 已知循环队列存储在数组 A 中，其下标范围为 0~m-1，头尾指针分别为 f 和 r，则将值为 x 的元素入队的操作序列是_____ _____。

3. 在单链表 L 的表头插入指针 S 所指结点的语句序列是_____。

4. 已知矩阵 A[0..9，0..9]的每个元素占 5 个存储单元，将其按行优先次序存储在起始地址为 1000 的连续的内存单元中，则元素 A[6,5]的地址为_____。

5. 对广义表 A=(a,(b,c,d))的运算 hcad(tail(A))的结果是_____。

6. 已知完全二叉树的第 7 层有 5 个叶子结点，则最多可能有_____个叶子结点。

7. 若二叉树的先序序列和后序序列相反，则该二叉树一定满足_____ _____。

8. 若无向图满足 _____ _____，则该图是树。

9. 对有序表 A[1..17]按二分查找方法进行查找，则查找长度为 5 的元素的下标从小到大依次是_____ _____。

10. 归并排序所需要的辅助空间的大小是_____。

四、解答下列各题（每小题 6 分，共 30 分）

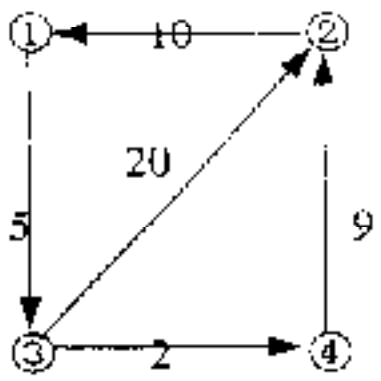
1. 已知一棵二叉树的先序、中序和后序序列如下，其中有些位置没有给出其值，请构造出该二叉树。

先序：A_CDEF_H_J

中序：C_EDA_GFI_

后序：C_ _BHGJI_ _

2. 已知图 G 如下，写出其邻接矩阵 A，按 Floyd 算法求解 $A^{(1)}$，并标明发生改变的元素。



$A=$ ( ) $A^{(1)}=$ ( )

3. 以下面数据序列为输入构造一棵平衡二叉树，要求画出最终结果，标明在插入哪个元素时出现不平衡以及作了何种调整，并求出在等概率情况下，查找成功时的平均查找长度。（给出计算公式即可）

100，75，85，90，120，135，110，180，95

4. 对下面数据表，写出采用快速排序算法排序的每一趟的结果，并标出数据移动情况。

（13　11　22　34　15　44　76　66　100　9　14　20　2　5　8）

5. 将下面数据表调整为大根堆。

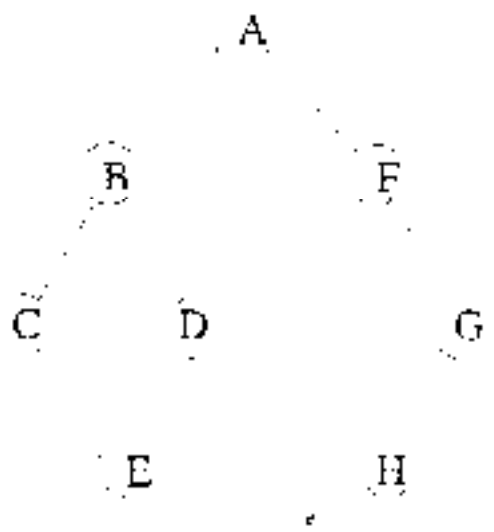
(10,20,50,30,40,60,90,100,80,72,13,91,18)

五、算法设计：分别写出求解下列问题的算法。（每小题 12 分，共 60 分）

1.已知带头结点的单链表 A 中各元素的值为整型数。设计算法将该链表分解为相同结构的 A 和 B 两个链表，使 A 表中存放原表中所有的奇数，B 表中存放原表中所有的偶数。

2.设计算法在左子树非空的中序线索二叉树 T 中插入值为 x 的结点，要求将其作为根结点的前趋插入到其左子树中，并维护其线索关系。

3. 二叉树的文本存储形式是指按先序次序给出各结点的值及其是否有左右孩子的标志，如果没有左右孩子，则相应的值为 1，否则为 0。例如，下面左图所示的二叉树的文本存储形式如右图所示。假设一棵非空二叉树的文本存储形式存储在一个足够大的数组 A 中，其最小下标为 1，每个元素是包含三个字段的记录（或结构），试编写算法由二叉树的文本存储形式构造出相应的二叉链表，并写出其适用形式和有关的类型描述。



Data	Ltag	Rtag
A	0	0
B	0	0
C	1	1
D	0	1
E	1	1
F	1	0
G	0	1
H	1	1

4.设计算法以判断有 n 个顶点的无向图 G 是否是一棵树，若是，则返回 true，否则返回 false。

（注：本算法中可以调用以下几个函数：

firstadj(G,V)——返回图 G 中顶点 V 的第一个邻接点的号码，若不存在，则返回 0；

nextadj(G,V,W)——返回图 G 中顶点 V 的邻接点中处于 W 之后的邻接点的号码，若不存在，则返回 0；

另外，如果涉及到栈或队列操作，可直接采用引用的形式，不必写出其实现形式）

5.已知顺序表有整型数组 data 和整型变量 listlen 两个分量，前者下标范围从 0 到 n，用于存放元素的值，其后者记录表中的实际元素个数，并已知 data 中的元素按从小到大的次序排列，设计算法以删除表中重复的元素，并要求时间复杂度为 O(n)。例如，对表(1,1,2,3,3,4,6,8,9,9) 执行的结果为(1,2,3,4,6,8,9)。

中国科学院合肥智能所 2003 年攻读硕士学位研究生入学考试试题

考试科目：《数据结构》

一、选择题：

1. (a)　　2. (c)　　3. (c)　　4. (d)　　5. (b)

二、判断题：

1. (√)　　2. (x)　　3. (x)　　4. (√)　　5. (√)

6. (√)　　7. (x)　　8. (√)　　9. (√)　　10. (x)

三、填空题：

1. (L^.next<>nil) and (L^.next^.next=nil)

2. r:=(r+1) mod m; A[r]:=x;

3. S^.next; L; L:=S;

4. 1325

5. (b, c, d)

6. 123

7. 仅有一个叶子结点

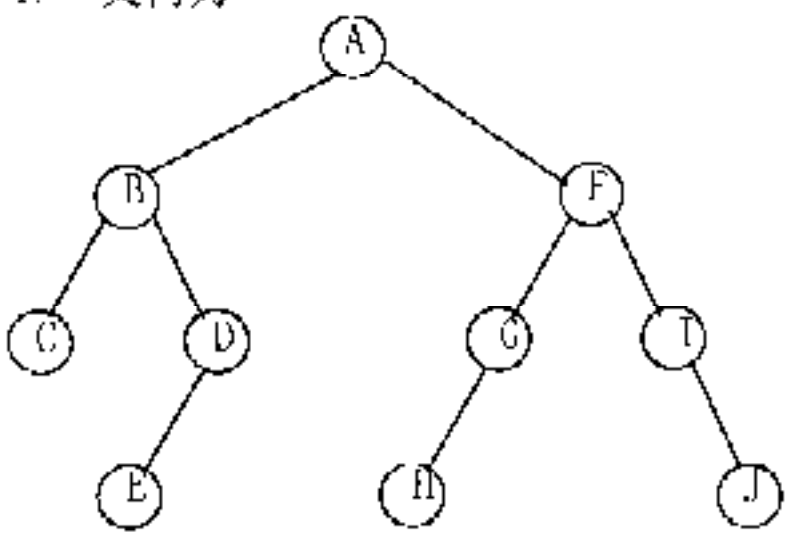
8. 有 n-1 条边的连通图

9. 8, 17

10. n

四、解答下列各题：

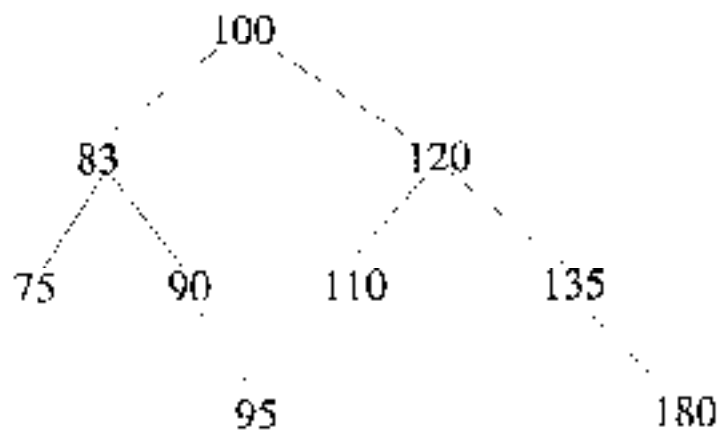
1. 又树为



2.

$$A=\begin{pmatrix} \infty & \infty & 5 & \infty \\ 10 & \infty & \infty & \infty \\ \infty & 20 & \infty & 2 \\ \infty & 9 & \infty & \infty \end{pmatrix} \qquad A^{[1]}=\begin{pmatrix} \infty & \infty & 5 & \infty \\ 10 & \infty & \boxed{15} & \infty \\ \infty & 20 & \infty & 2 \\ \infty & 9 & \infty & \infty \end{pmatrix}$$

3. 平衡二叉树如下：



其中在插入 85 时作 RL 型调整，在插入 135 时作 RR 型调整。

平均查找长度 Asl=(1*1+2*2+3*4+4*2)/9=25/9

4. 快速排序

（13 11 22 34 15 44 76 66 100 9 14 20 2 5 8）

（8 11 22 5 15 2 9） 13 (100 66 14 20 76 44 34)

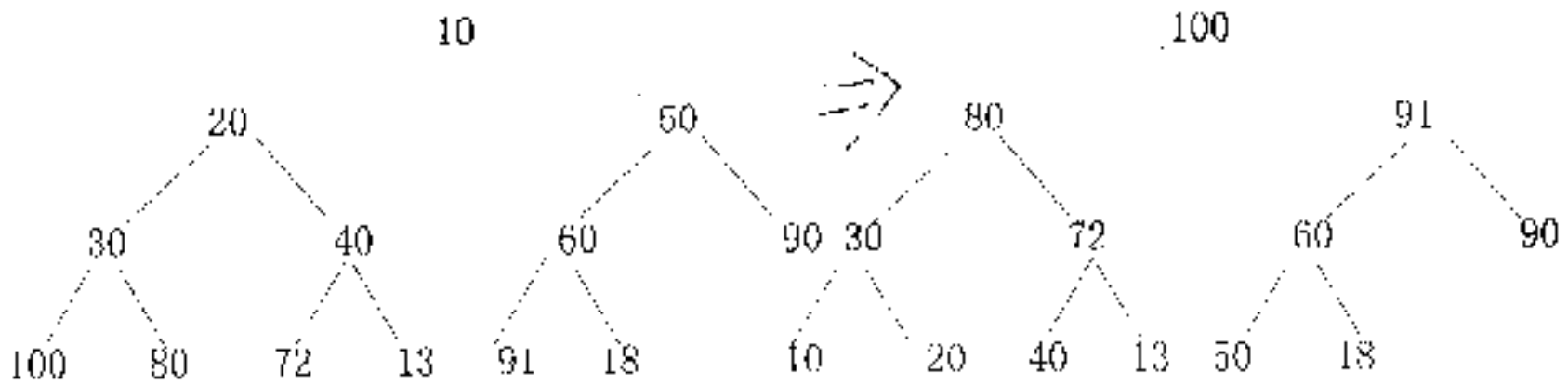
（2 5） 8 (22 15 11 9) (34 66 14 20 76 44) 100

2 (5) (9 15 11) 22 (20 14) 34 (66 76 44)

9 (15 11) (14) 20 (44) 66 (76)

(11) 15

5. 将下面数据表调整为大根堆。



五、算法设计

1. 算法如下：

```
procedure  Disp(var A,B:Link);
   begin
     new(B); R:=B;
     P1:=L; P2:=L^.next;
     while  (P2 <>nil)  do
        if odd(P2^.data)
        then begin P1:=P2; P2:=P2^.next; end
```

```
        else begin P1^.next:-P2^.next;
                   R^.next:-P2; R:=P2;
                   P2:=P1^.next
             end;
       R^.next:=nil;
    end;
```

2.

```
procedure  Insert(T:bitre;x:datatype);
 begin
    new(S); S^.data:-x;
    P:=T^.lchild;
    while (P^.rtag=0) do P:=P^.rchild          {往右下搜索根的前趋结点}
    S^.Lchild: P; S^.ltag:=1;
    S^.rchild:-T; S^.rtag:=1;
    P^.rchild:=S; P^.rtag:=0;
End;
```

3.

```
procedure Create(var T:bitre);     //全局变量 I 的初值为 1
     var Cur I:integer;
     begin
       curI:=I; I:=I+1;
       new(T); T^.data:=A[CurI];
       if A[CurI].ltag=1 then T^.lchild:=nil
              else begincreate(T^.Lchild);
       if A[Curi].rtag 1 then T^.rchild:-nil
              create(T^.rchild);
    End;
```

4.

```
function judge(var g:datagraph):boolean;
  begin   setnull(Q);
    for I:=1 to n do
       visited[i]:=false;
    visited[1]:=true; write(1); Enqueue(Q,1);  Enum:=0; Vnum:=1;
    while not empty(Q) do
    begin  v:-outquene(Q);  w: firstadj(G,v);
     while  w<>0 do
        begin    Enum:=Enum+1;
           if not visited[w] then
             begin  visited[w]:=true;   Enqueue(Q,w); Vnum:=Vnum+1 end;
```

```
            w: nextadj(G, v, w);
        end;
        return((Enum=2*n-2)and(vnum=n))
    end;
```

5. 本算法采用C语言描述如下

```
void delete (seqlist *L)
{int current=0, newlast=0, j;
 while (current<L->listlen-1)          // current 所指后面还有元素时
   { j=current+1;              //以 j 指示搜索与 current 所指不同的元素
     while (J<L->listlen && L->data[J]== L->data[Current]) J++;
     if (J<L->listlen)
       { L->data[++newlast]== L->data[J];     //保存所搜索到的不同元素到前面区域
         Current=j;                           //继续新的元素的搜索
       }
     else Current=j;
   }
   L->listlen=newlast+1;                                  //重新设定表长度
}
```