北京交通大学

2009 －2010 学年第二学期期末考试试题

课程名称： 数据结构（A） 出题教师：李向前、赵帅峰、徐薇

专业:_____ 班级:_____ 姓名:_____ 学号:_____

题号	一	二	三	四	五	六	七	总分
得分								
阅卷人								

（注意：所有答案均写在答题纸上，否则无效）

一、 填空题(每题 1 分，共 20 分)

1．一个算法是由控制结构和_____构成的，算法的时间取决于两者的综合效果。

2．在表长为 n 的顺序表中，共有_____个插入位置，如果要在第 i 个数据元素之前插入一个新的元素，要移动_____个数据元素。

3．循环队列存储在数组 A[0..7]中，其头、尾指针分别是 front=1 和 rear=0，则队列中的元素个数为 _____。

4. 若串 S=‘MBNDEFLGEFK’,则 StrDelete(&S,3,4)的结果是_____。

5. 广义表(c,((a,m)),d,(e,g),(((i,j)),k))的长度是_____ ,深度是_____。

6. 稀疏矩阵中的元素以三元组顺序表来表示，则矩阵中的元素（5，7，9）在相应转置矩阵中的表示为_____。

7.一棵完全二叉树共有 800 个结点，其叶子结点有_____个。

8.在哈夫曼树中有 20 个叶子结点，度为 1 的结点有_____个。

9.在中序线索二叉树中，某结点有左子树，但没有右子树，则该结点的前驱是_____。

10.对森林进行中序遍历，相当于对森林中的每一棵树从左到右进行_____遍历。

11.对有 n 个顶点的有向完全图，用邻接表表示，则邻接表中共有_____个弧结点 。

12.对于有 e 条边的无向图，所有顶点的度的和为_____。

13．对于_____网，可以求得关键路径；对于_____网，可以进行拓扑排序。

14.有 n 个顶点的有向强连通图至少需要_____条弧。

15.在等概率的情况下，对有 15 个元素的顺序表进行折半查找的平均查找长度为_____。

16．在有 10 个结点的完全二叉树上，所有结点的平衡因子之和为_____。

17．一棵 7 阶的 B_树中，非终端结点中的关键字个数最少为_____个。

18．稳定的排序方法有_____、_____、_____。

19．对关键字序列（34，23，47，89，18，42）进行一趟快速排序的结果是_____。

20．若二叉树有 m 个叶子结点，则二叉树的高度至少为_____。

二、 选择题(每题 2 分，共 20 分)

1、一个单链表有头结点，它的头指针是 L,通过如下（ ）的表示可以指向第二个数据元素。

A．L B．L->next C．L->next->next D．L->next->data

2、一组字符 abcde 按顺序先进入一个栈，然后出栈，再按顺序进入一个队列再出队列，最后得到字符序列为（ ）。

A.abcde B.aebdc C.edcba D.eadbc

3、模式串 T=‘abcaabbcabcaabdaa’，该模式串的 next 数组值及 nextval 数组值为（ ）。

A.01112231123456712 和 01100111011001702
B.01112231124567112 和 01102131011021701
C.01112121124567112 和 01100111011001702
D.01112231123456712 和 01102131011021702

4、已知广义表 LS=((a,b,c)，(d,e,f))，运用 head 和 tail 函数取出 LS 中原子 b 的运算是（ ）。

A.head(tail (LS)) B.tail(head(LS))
C.head(tail(head(tail(LS)))) D.head(tail((head(LS))))

5、二叉树中度为 2 的结点有 5 个，度为 1 的结点有 4 个，则结点总数为（ ）。

A.13 B.14 C.15 D.16

6、某二叉树中序序列为 ABCDEFG，后序序列为 BDCAFGE，则先序序列为（ ）。

A.EGFACDB B.EACBDGF C.EAGCFBD D.上面的都不对

7、深度为 10 的二叉树拥有的最少结点数为（ ）。

A. 1023 B. 512 C. 511 D. 以上都不对

8、对于线性表(27,25,72,50,42,32,90)进行散列存储时，若选用 H(K)=K % 9 作为散列函数，用线性探测再散列来解决冲突构造散列表，则散列地址为 0 的元素有（ ）个。

A.0 B.1 C.2 D.3

9、对以下关键字序列用快速排序法进行排序，速度最慢的是（ ）。

A.（4,26,7,15,19,43） B. (32,9,6,22,41,32)
C.(28,53,75,81,41,56) D. (7,9,10,18,20,21,29)

10、任何一棵二叉树的叶子结点在先序、中序和后序遍历序列中的相对次序（ ）。

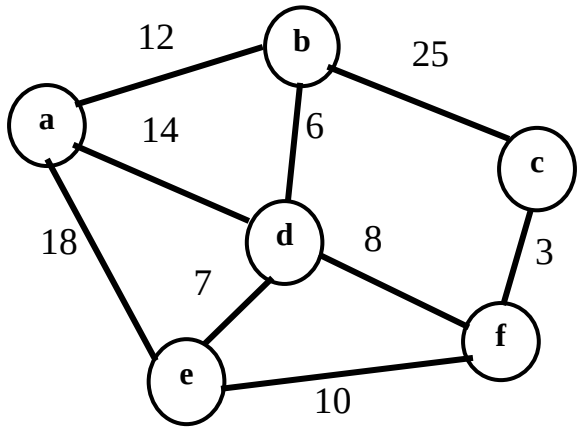
A．不变 B．发生改变 C．不能确定 D．以上都不对

三、 判断是非题(每题 1 分，共 10 分)

( )1. 非空循环链表中，任一结点的指针均不空。
( )2. 广义表中的元素或者是一个不可分割的原子，或者是一个非空的广义表。
( )3. 已知一棵非空二叉树的两个遍历序列，一定能构造出该二叉树。
( )4. 完全二叉树中，若一个结点没有左孩子，则它必是叶子结点。
( )5. 相比于 Kruskal 算法，Prim 算法更适合于求边稠密的无向网的最小代价生成树。
( )6. 有向图和无向图均可以采用数组表示法和十字链表作为其物理存贮结构。
( )7. 在一个有向图的邻接表或逆邻接表中，如果某个顶点的链表为空，则该顶点的度不一定为零。
( )8. 哈希函数越复杂越好，因为这样随机性好，冲突概率小。
( )9. 若把堆看成是一颗完全二叉树，则该树一定是一棵二叉排序树.
( )10. 由于希尔排序的最后一趟与直接插入排序过程相同，因此前者一定比后者花费的时间更多。

四、 画图题(每题 5 分，共 20 分)

1、 请使用 Prim 方法，求带权无向图的最小生成树及其代价（以 a 为起始点，给出过程）。



2、 已知一组关键字序列为：(46, 74, 18, 53, 14, 26, 40, 38, 86, 65)，请构造二叉排序树和平衡二叉树。

3、 表长度为 7，哈希函数为 H(key)=（3key） MOD 7，用线性探测再散列法处理冲突，画出对于给定的如下一组关键字{18，51，31，3，25，9 }所得到的哈希表，并求等概率情况下查找成功时的平均查找长度。

4、已知二叉树的先序遍历序列： e a d c b j f g l h
和中序遍历序列： a c b d j e f l g h

a. 画出该二叉树；
b. 画出该二叉树的中序线索二叉链表，用虚线表示线索。

五、 程序填空题(每题 2 分，共 10 分)

1、下列算法实现堆排序，请把程序补齐。

```
typedef  SqList  HeapType;   //堆采用顺序表存储表示
```

```
void HeapAdjust (int &R[], int s, int m)
{  // 已知 R[s..m]中记录的关键字除 R[s] 之外均满足堆的特征
   //本函数自上而下调整 R[s] 的关键字，使 R[s..m] 也成为一个大顶堆
rc = R[s];                              // 暂存 R[s]
for ( j=2*s; j<=m; _____)
{ if ( j<m && _____ )  ++j;
  if ( rc.key >= R[j].key )  break;
  R[s] = R[j];   s = j;    }
 _____;
}

void HeapSort(HeapType  &H) //对顺序表 H 进行堆排序
{  for (i=H.length/2; i>0; --i)
     _____ ;
   for (i=H.length; i>1; --i)
    { H.r[1]←→H.r[i];
     _____ ;
    }
}
```

六、 算法说明题(每题 10 分，共 10 分)

1. 说明下列算法的功能，并写出算法执行的结果。

```
typedef struct ArcNode {
    int  adjvex;
    struct ArcNode  *nextarc;
} ArcNode;

typedef struct VNode {
    int  vertex;
    ArcNode  *firstarc;
  } VNode, AdjList[6];

int n=6; e=8;
void unknown1 (AdjList &G)
{
  int arcs[][2] = {{0,1},{1,2},{1,4},{2,3},{3,1},{3,4},{3,5},{5,0}};
  for (i=0;i<n;i++)
    {  G[i].vertex=i;
       G[i].firstarc=null;}
  for (i=0;i<e;i++)
    { p=(ArcNode*)malloc(sizeof(ArcNode));
      p->adjvex= arcs[i][1];
      p->nextarc=G[arcs[i][0]].firstarc;
      G[arcs[i][0]].firstarc=p;
    }
 }
```

```
void unknown2 (AdjList G)
{ for(i=0;i<n;i++)
  {  num=0;
     for(j=0;j<n;j++)
     if(i!=j)
      { p=G[j].firstarc;
        while(p)
          {if(p->adjvex==i) num++;p=p->nextarc;}
       }
     printf(“\n%d%d”,G[i].vertex,num);
   }
}

main ()
{ AdjList G;
  unknown1 (G);
  unknown2 (G);
}
```

七、 算法设计题(每题10分，共10分)

1. 已知一棵完全二叉树存放于一个一维数组T[10]中，试设计一个算法，从T[0]开始顺序读出各结点的值，建立该二叉树的二叉链表表示。

```
DataType T[10];
typedef struct BiTNode {
    DataType      data;
    struct BiTNode  *lchild, *rchild;
} BiTNode, *BiTree;
```

北 京 交 通 大 学

2009 －2010 学年第二学期期末考试试题

《数据结构》答案

一、填空题（每题 1 分）

1. 原操作
2. n+1 n-i+1
3. 7
4. ‘NDEF’
5. 5 4
6. (7,5,9)
7. 400
8. 0
9. 其左子树中最右下结点
10. 后根
11. n(n-1)
12. 2e
13. AOE AOV
14. n
15. 49/15
16. 2
17. 3
18. 直接插入、起泡、基数、归并
19. (18,23,34,89,47,42)
20. $\lceil \log_2 m \rceil + 1$

二、选择题（每题 2 分）

1、C
2、C
3、D
4、D
5、C
6、B
7、D
8、B
9、D
10、A

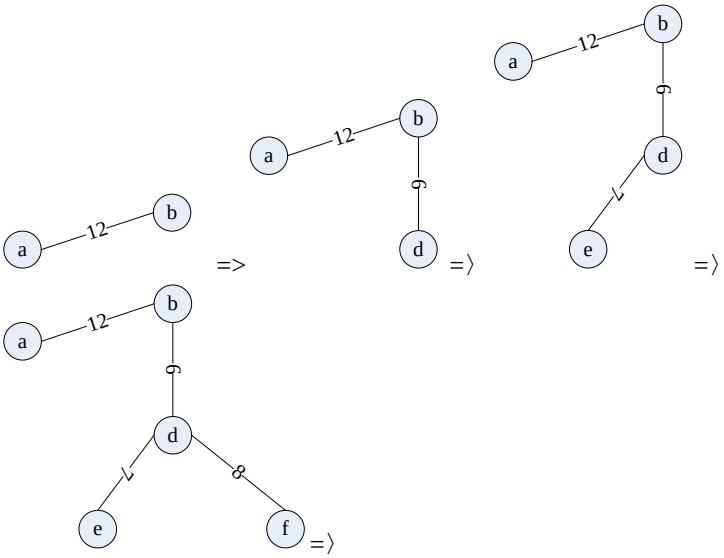
三、判断是非题（每题 1 分）

1、√
2、X
3、X
4、√
5、√
6、X
7、√
8、X
9、X

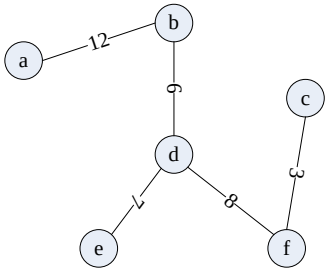
10、X

四、画图题

1.



1 分　　2 分

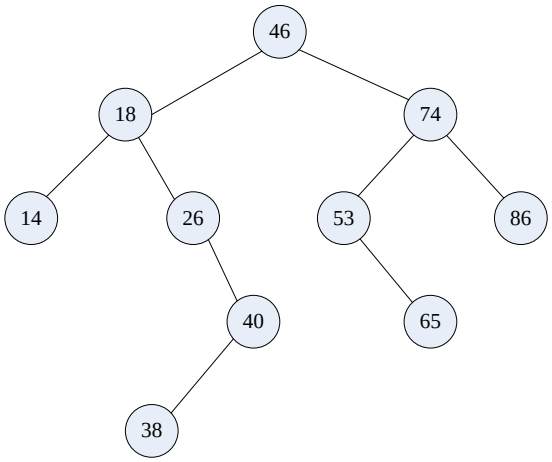


1 分

代价：12 + 6 + 7 + 8 + 3 = 36.　　……1 分

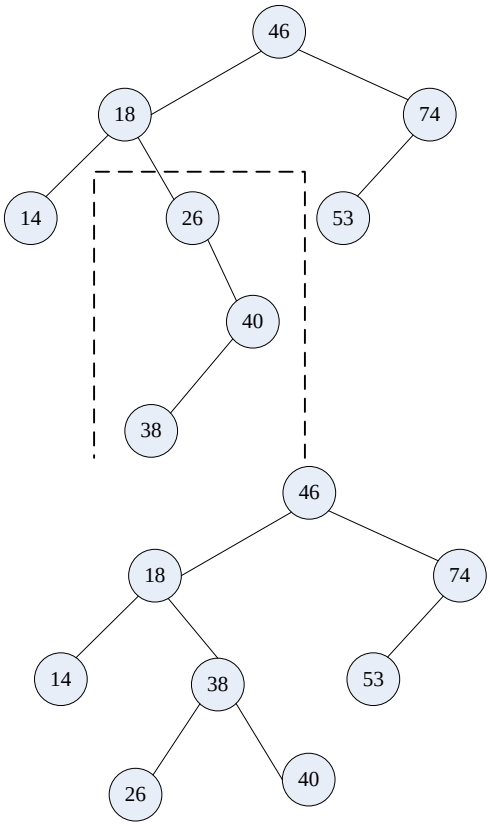
2.

二叉排序树:

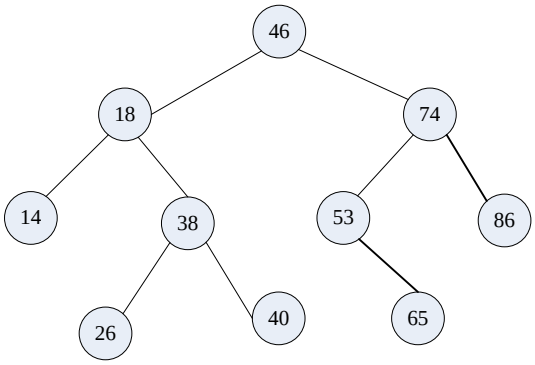


2 分

平衡二叉树：当增加 38 时，失去平衡。



=> 2 分



=> 1 分

3．根据 HASH(key) = 3key MOD 7,函数，求到的地址如下：……1 分

Key	18	51	31	3	25	9	
HASH(key)	5	6	2	2	5	6	

采用线性探测再散列法处理冲突，得到的 HASH 表如下：

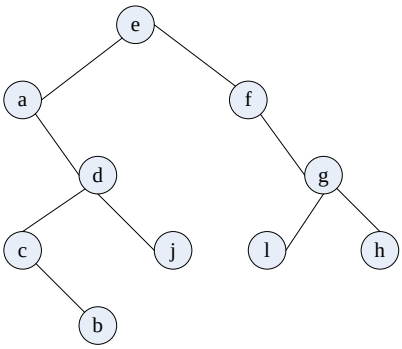
地址	0	1	2	3	4	5	6
Key	25	9	31	3		18	51

放对 18,51,31，得到 1 分；

放对 3，25，9 共得 2 分

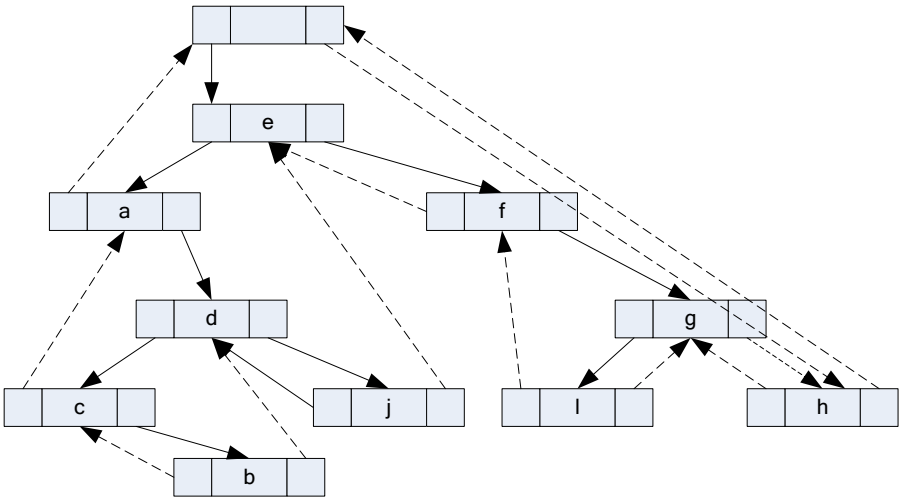
平均查找长度：(1+1+1+2+3+3)/6 = 11/6. ……..1 分

4.、
该二叉树为：



2 分

中序线索二叉链表：



头结点：1 分
其他：2 分。

五、程序填空题（每空 2 分）
1、j=j*2
2、R[j].key＜R[j+1].key
3、R[s] = rc
4、HeapAdjust(H,i,H.length)
5、HeapAdjust(H,1,i-1)

六、算法说明题

算法功能：建立有向图的邻接表存储结构（3 分），求各顶点的入度（2 分）
运行结果：（5 分）

```
0   1
1   2
2   1
3   1
4   2
5   1
```

八、　算法设计题

```
void CreateBiTree(DataType T[];int n;int i;BiTree  &ptr)
 {  if (i>=n)  ptr = NULL;                          ---------- 2 分
    else {
       ptr = (BiTNode *)malloc(sizeof(BiTNode));---------- 1 分
       ptr->data = T[i];                            ---------- 1 分
       CreateBiTree(T,n,2*i+1;ptr->lchild);         ---------- 2 分
       CreateBiTree(T,n,2*i+2;ptr->rchild); }       ---------- 2 分
      }
 }

main( )                                             ----------2 分
{  BiTree root=null;
   n=10;
   CreateBiTree(T,n,0;root);
}
```

北京交通大学

2009 －2010 学年第二学期期末考试试题

课程名称： 数据结构（B） 出题教师：李向前、赵帅峰、徐薇

专业:_____ 班级:_____ 姓名:_____ 学号:_____

题号	一	二	三	四	五	六	七	总分
得分								
阅卷人								

（注意：所有答案均写在答题纸上，否则无效）

一、 填空题(每题 1 分，共 20 分)

1．若额外空间相对于输入数据量来说是常数，则称此算法为_____。

2．在表长为 n 的顺序表中，共有_____个删除位置，如果删除第 i 个数据元素，要移动_____个数据元素。

3．链队列的头指针和尾指针分别是 Q.front 和 Q.rear，则链队列为空的判定条件是_____。

4. 若串 S=‘GUSEKOLWJFI’,则 SubString(&Sub,S,5,3)的结果是_____。

5. 广义表 A=(b,A)的长度是_____ ,深度是_____。

6. 一个 n 阶对称矩阵可以存储在长度为_____的一维数组中。

7.有 2048 个结点的完全二叉树的深度为_____。

8.在哈夫曼树中有 20 个叶子结点，则结点总数为_____。

9.在中序线索二叉树中，某结点有右子树，但没有左子树，则该结点的后继是_____。

10.对森林进行先序遍历，相当于对森林中的每一棵树从左到右进行_____遍历。

11.对于有 n 个顶点的无向完全图，用邻接表表示，则邻接表中共有_____个边结点 。

12. n 个顶点的连通图的生成树有且仅有_____条边。

13．一个有 e 条弧的有向图的十字链表中共有_____个弧结点。

14._____算法适合于求边稀疏的网的最小生成树。

15.在等概率的情况下，顺序查找的平均查找长度为_____。

16. 在有 13 个结点的完全二叉树上，所有叶子结点的平衡因子之和为_____。

17. 一棵 6 阶的 B_树中，非终端结点中的关键字个数最多为_____个。

18. 平均排序时间为 O(nlogn)的排序方法有_____、_____、_____。

19. 15 个关键字进行 2 路归并排序，要得到一个有序序列，一共要进行_____趟归并。

20. 对 n 个数据元素的有序表进行折半查找，在查找成功时和给定值进行比较的关键字个数至多为_____。

二、 选择题(每题 2 分，共 20 分)

1、带头结点的单链表 head 为空的判断条件是（　　）。

A. head= =NULL　　B. head->next= =NULL

C. head->next= =head　　D. head!=NULL

2、一个栈的输入序列为 12345，则下列序列中不可能出现的输出序列是(　　）。

A. 23415　　B. 54132　　C. 23145　　D. 15432

3、AVL 树是一种平衡的二叉排序树，树中任一结点的(　　）。

A.左、右子树的高度均相同　　B.左、右子树高度差的绝对值不超过 1

C.左子树的高度均大于右子树的高度　　D.左子树的高度均小于右子树的高度

4、已知广义表 LS=((((),a)),(d,(c)),((e)),b),则它的长度和深度分别为___。

A.5 和 4　　B.4 和 4　　C.3 和 4　　D.4 和 3

5、二叉树中度为 2 的结点有 6 个，度为 1 的结点有 5 个，则结点总数为（　　）。

A.15　　B.16　　C.17　　D.18

6、排序方法中，关键字比较的次数与记录的初始排列无关的是（　　）。

A. Shell 排序　　B. 快速排序　　C. 直接插入排序　　D. 简单选择排序

7、深度为 10 的二叉树拥有的最多结点数为（　　）。

A. 1023　　B. 512　　C. 511　　D. 以上都不对

8、图的广度优先搜索类似于二叉树的(　　)遍历。

A. 先根　　B. 中根　　C. 后根　　D. 层次

9、关于完全二叉树，下列描述错误的是 ___。

A.完全二叉树可以用来构造堆进行排序

B.完全二叉树相当于具有相同深度的满二叉树缺少最后 n（n>=0）个连续的结点

C.赫夫曼树不一定是完全二叉树

D.折半查找的判定树就是完全二叉树

10、起泡排序在最好情况下的时间复杂度为____。

A. O(logn)　　B. O(n)　　C. O(n*logn)　　D. $O(n^2)$

三、 判断是非题(每题 1 分，共 10 分)

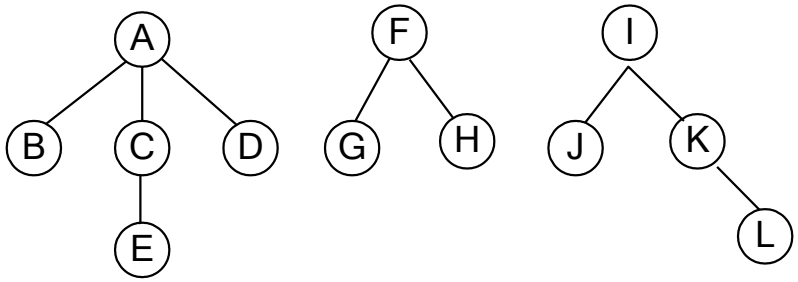
(　)1. 在单链表中，要访问某个结点，只要知道该结点的指针即可，因此，单链表是一种随机存取结构。

(　)2. 广义表的取表尾运算，其结果通常是个表，但有时也可能是单个元素值。

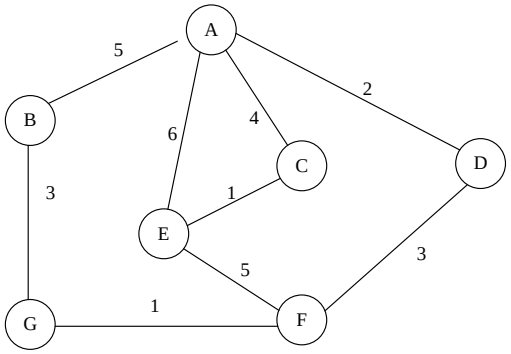
( )3. 二叉树按某种顺序线索化之后，任一个结点均有指向其前驱结点或者后继结点的线索。
( )4. 完全二叉树中，若一个结点没有左孩子，则它必是树叶。
( )5. 对于有 n 个结点的二叉树，其高度为 $\log_2 n$.
( )6. 堆肯定是一棵平衡二叉树。
( )7. 用邻接矩阵存贮一个图时，在不考虑压缩存储的情况下，所占用的存储空间大小与图中结点个数有关，而与图的边数无关。
( )8. 在表示某工程的 AOE 网中，加速其关键路径上的任意关键活动，均可缩短整个工程的完成时间。
( )9. 折半查找法的查找速度一定比顺序查找快。
( )10. 平衡二叉树中，若某个结点的左右孩子的平衡因子为零，则该结点的平衡因子一定是零。

四、 画图题(每题 5 分，共 20 分)

1、设待排序的关键码序列为{12，2，16，30，28，10，16*，20，6，18}，要求排序结束后关键码非递减有序排列，请写出建立初始堆的过程。
2、已知一组关键字序列为：(60，70，50， 40，30， 55，58)，按顺序输入，画出其构成的平衡二叉树，给出过程。
3、树采用孩子兄弟表示法，二叉树采用二叉链表表示法，二叉树和森林可以相互转换，请把下图表示的森林转换成二叉树。



4、考虑下图
（1）给出该图的邻接表；
（2）根据（1）中给出的邻接表，从顶点 A 出发，求它的深度优先遍历序列；
（3）根据普里姆（Prim）算法，求它的最小生成树。



五、 程序填空题(每题 2 分，共 10 分)

下列算法实现图的广度优先遍历，请把程序补齐。

```
void BFSTraverse(Graph G) // 对图 G 作广度优先遍历
{
  for (v=0; v<G.vexnum; ++v)  visited[v] = _____  ;
  InitQueue(Q);
  for ( v=0;  v<G.vexnum;  ++v )
    if ( !visited[v])
      {  visited[v] = _____ ; printf(v);
         EnQueue(Q, v);
         while ( _____  )
          { DeQueue(Q, u);
            for(w=FirstAdjVex(G, u); w!=0;  w=NextAdjVex(G,u,w))
              if (  _____ )
                   { visited[w]=TRUE; printf(w);
                     _____ ; }
           } // while
       }
} // BFSTraverse
```

六、 算法说明题(每题 10 分，共 10 分)

1. 说明下列算法的功能，并写出算法执行的结果。

```
typedef struct  LNode {
        int      data;
        struct Lnode   *next;
     } LNode, *LinkList;
LinkList  L;

void  unknown1 (LinkList &L)
{  int  a[10]={1,2,3,4,5,6,7,8,9,10};
   Int n=10;
   L = (LinkList) malloc (sizeof (struct LNode));
   L->next = NULL;
   for (i = n-1; i >= 0; --i)
     {  p = (LinkList) malloc (sizeof (struct LNode));
       p->data=a[i];
       p->next = L->next; L->next = p;
     }
}
```

```
int  unknown2 (LinkList L, int k)
{  p=L; q=L->next;   i=1;
   while(q)
    { q=q->next;  i++;
      if(i>k)  p=p->next; }
   if (p==L) return 0;
    else
    { printf(“\n %d“,p->data);
     return 1; }
 }

main()
{ LinkList L;
  L=unknown1(L);
  p=L->next;
  while(p)
  { printf(“%d”,p->data);
   p=p->next; }
  unknown2(L,3);
}
```

七、 算法设计题(每题 10 分，共 10 分)

1、设有两个有序的单链表，一为升序，一为降序，试编写一个程序，将这两个链表合并为一个有序链表。

北 京 交 通 大 学

2009 －2010 学年第二学期期末考试试题

《数据结构》答案

一、填空题（每题 1 分）

1. 原地工作
2. n n-i
3. Q.front = Q.rear
4. Sub =‘KOL’
5. 2 无穷大
6. n(n+1)/2
7. 12
8. 39
9. 其右子树中最左下结点
10. 先根
11. n(n-1)
12. n-1
13. e
14. 克鲁斯卡尔
15. (n+1)/2
16. 0
17. 5
18. 快速排序、归并排序、堆排序
19. 4
20. $\log_2 n\rfloor + 1$

二、选择题（每题 2 分）

1、B
2、B
3、B
4、B
5、D
6、D
7、A
8、D
9、D
10、B

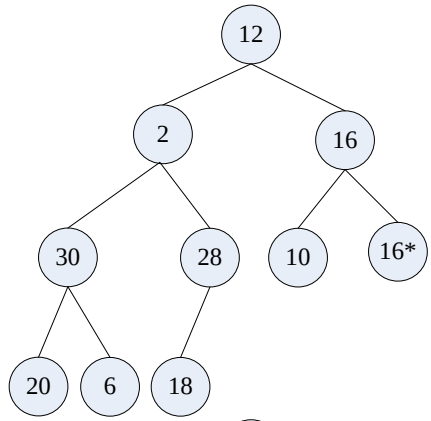
三、判断是非题（每题 1 分）

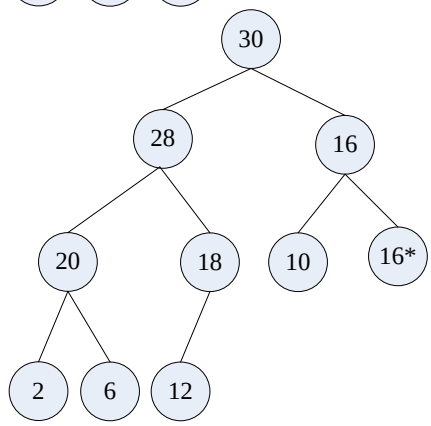
1、X
2、X
3、X
4、 √
5、X
6、X
7、 √
8、X
9、X

10、X

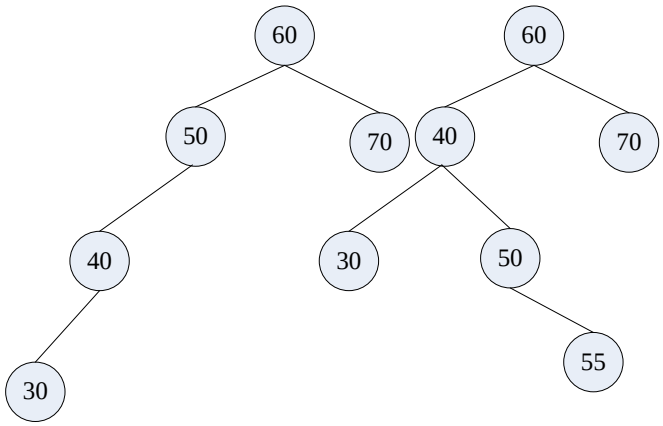
四、画图题

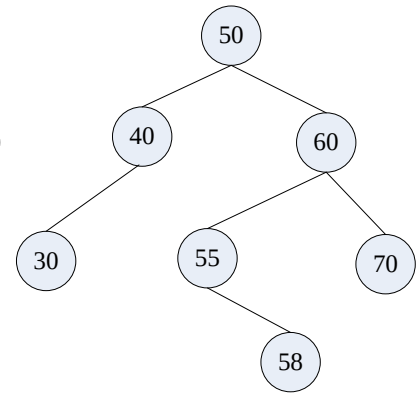
1、需要建立大顶堆排序



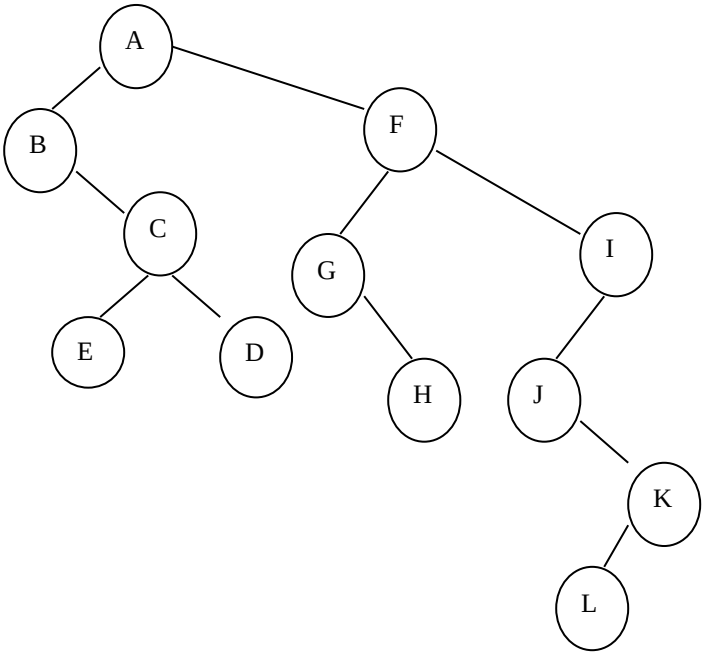


2、

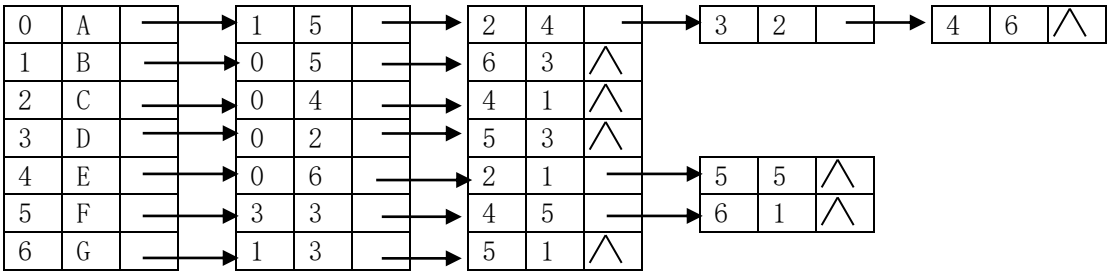




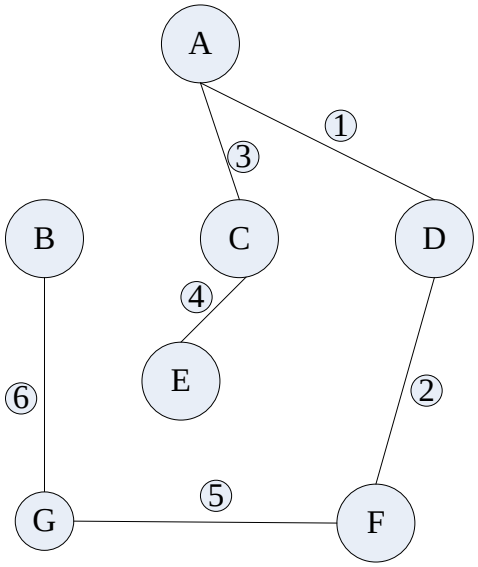
3、



4、



深度优先搜索序列为 ABGFDEC



最小生成树如图所示，它的代价为 14

五、程序填空题（每空 2 分）

1、FALSE

2、TRUE

3、!QueueEmpty(Q)

4、! visited[w]

5、EnQueue(Q, w)

六、算法说明题

算法功能：建立带头结点的单链表并输出，查找链表中倒数第 k 个位置上的结点，查找成功，输出该结点的 data 值，并返回 1；否则，只返回 0。(5 分)

运行结果：1 2 3 4 5 6 7 8 9 10

8 (5 分)

七、算法设计题

```
Typedef struct  LNode {
        ElemType      data;
        struct Lnode   *next;
     } LNode, *LinkList;
LinkList  L;      (2 分)

LinkList Union(Linklist &La,Lb)
{    //把升序链表 La 和降序链表 Lb 合并为升序链表 La，两个链表均带头结点
   p=Lb->next; Lb->next=Null;
   while(p)
       { s=p->next;
          p->next=Lb->next; Lb->next =p;p=s} //Lb 逆置          (4 分)
   p=La->next;  q=Lb->next; pre=La;
   while( p & q )
    if (p->data<q->data) {pre->next=p;pre=p;p=p->next}
     else {pre->next=q;pre=q;q=q->next}
   if ( p ) pre->next=p;
   else pre->next=q;
   return La;
}                                                          (4 分)
```

一、单项选择(每题 2 分，共 20 分)

1、链表不具备的特点是（ ）

A．可以随机访问任一结点　　B．插入删除不需要移动元素

C．不必事先估计存储空间　　D．所需空间与其长度成正比

2、任何一棵二叉树的叶子结点在先序、中序和后序遍历序列中的相对次序（ ）

A．不发生变化　B．发生变化　C．不能确定　D．以上都不对

3、设有 999 个无序的元素，希望用最快速度挑出其中前 5 个最大的元素，采用下述哪种方法最好。（ ）

A.快速排序　　B.堆排序　　C.基数排序　　D. Shell 排序

4、某二叉树的先序序列和后序序列正好相反，则该二叉树一定是（ ）

A．空或只有一个结点　　B．高度等于其结点数

C．任一结点无左孩子　　D．任一结点无右孩子

5、排序方法中，关键字比较的次数与记录的初始排列无关的是（ ）。

A．Shell 排序　　B．快速排序

C．直接插入排序　　D．简单选择排序

6、字符串’ababaabab’的 next[j] 函数的值为（ ）。

A．（0，1，0，1，0，4，1，0，1）　　B.（0，1，1，2，3，4，2，3，4）

C．（0，1，0，1，0，0，0，1，1）　　D.（0，1，0，1，0，1，0，1，1）

7、在遍历中序线索二叉树时，某结点既有左子树又有右子树，那么它的前驱是其（ ）.

A.右子树中最左下的结点　　B.右子树中最右下的结点

C.左子树中最左下的结点　　D.左子树中最右下的结点

8、对某个无向图的邻接矩阵来说，下列哪种说法是正确的。（ ）

A．第 i 行上的非零元素个数和第 i 列的非零元素个数一定相等

B．矩阵中的非零元素个数等于图中的边数

C．第 i 行上和第 i 列上非零元素总数等于顶点 Vi 的度数

D．矩阵中非全零行的行数等于图中的顶点数

9、图的广度优先搜索类似于二叉树的( )遍历。

A．先根　　B．中根　　C．后根　　D．层次

10、广义表 A((x，(a ,b)), (x , (a , b) , y))，则运算 Head(Head(Tail(A)))为（ ）。

A．x　　B．（a , b）　　C．（x，（a ,b ））　　D． A

第一题答题处

1	2	3	4	5	6	7	8	9	10

二、填空(每题 1 分，共 10 分)

1．由一个长度为 7 的有序表，按二分查找法对该表进行查找，在表内各元素等概率情况下，查找成功的平均查找长度是_____。

2．一个 N 阶对称矩阵，矩阵元为 A_{ij},将其下三角部分以行序为主序存放在一维数组 M[0,n(n+1)/2-1]中，设矩阵最左上角矩阵元为 A_{00}，则 M[29]对应的矩阵元为_____。

3．一棵 5 阶的 B_树中某个结点的关键字个数最多为_____个。

4．一棵完全二叉树的第六层上有 9 个结点，则结点总数是_____。

5. 在一个具有 n 个顶点的有向完全图中，包含有 _____条边。
6. 在一棵二叉树中，假设度为 2 的结点有 8 个，度为 1 的结点有 7 个，则叶子结点数有_____个。
7. 在含有 n 个结点的二叉树的二叉链表中共有个_____空链域。
8. 设正文串长度为 a，模式串长度为 b，则串匹配的 KMP 算法的时间复杂度为_____。
9. 有 n 个顶点的无向连通图至少需要_____条边。
10. 一个无向图中有 e 条边，那么它的邻接表中有_____个边结点。

第二题答题处

1	2	3	4	5	6	7	8	9	10

三、判断对错(每题 1 分，共 10 分)

1. （ ）算法的衡量指标主要在于算法的时间复杂度和空间复杂度。
2. （ ）和顺序存贮结构不同，线性表的链式存贮结构更适用于完成插入和删除运算，因此，链式存贮结构比顺序存贮结构更好更常用。
3. （ ）循环队列满是指为队列分配的存贮空间已经全部存放了数据元素。
4. （ ）具有 n 个叶子节点的哈夫曼树的结点总数为 2n-1。
5. （ ）树和二叉树可以相互转换。
6. （ ）树的度是指根节点的最大子树数。
7. （ ）一棵 7 阶的 B-树，所有非终端结点最多有 7 棵子树，最少有 4 棵子树。
8. （ ）对于 AOE 网络，加速任一关键活动都能使整个工程提前完成。
9. （ ）在一个有向图的邻接表或逆邻接表中，如果某个顶点的链表为空，则该顶点的度一定为零。
10. （ ）查找表是同一类型的数据元素构成的集合，因此对查找表进行查找的各种算法与查找表的存贮结构无关。

四、作图/解答（每题 5 分，共 20 分）

1、若图的邻接矩阵如下图示， 用图示方法， 画出其邻接表的存贮结构。

3	∞	∞	2
∞	∞	8	∞
5	3	∞	∞
∞	∞	∞	∞

2、已知一组关键字序列为：(42, 84, 18, 50, 14, 26, 48, 38, 86, 65)，

1 ）请画出其构成的二叉排序树，

2 ）请画出其构成的平衡二叉树。

3、已知一组关键字序列为（12，51，8，22，26，80，11，16，54，41），其散列地址空间为[0,…,12]，若 Hash 函数定义为：H(key) = key MOD 13，采用线性探查法处理冲突，请给出它们对应的散列表。

4、已知一组关键字序列为：(46, 74, 18, 53, 14, 26, 40, 38, 86, 65)，若采用堆排序方法进行排序，请构造初始堆，要求：最后输出结果按关键字非递减有序排列。

五、证明题和问答题（每题 5 分，共 10 分）

1、当三叉树中只有度为 3 和 0 的结点时，证明 $n_3=(n_0-1)/2$，其中 n_3 是度为 3 的结点数，n_0 是度为 0 的结点数。

2、有 n 个互不重复的正整数型关键字进行堆排序，在构造的初始堆中，叶子结点数有多少？结点总数有多少？分支结点数有多少？

六、算法阅读（每题 5 分，共 10 分）

1、阅读下列程序，说明该算法完成的功能，写出程序执行后的结果。

```
#include <stdio.h>
#include <malloc.h>
typedef struct node { char data;  struct node *next; }LINKLIST;
```

```
char strings[8]= ’ datastr$’ ;

LINKLIST *invertlink(LINKLIST *head)
{ LINKLIST *p, *q, *r;
  q = NULL;         p = head;
  while(p != NULL)
     {r = q;   q = p;    p = p->next;   q->next = r;}
  return q;
}

LINKLIST *creatlink_nohead_head(LINKLIST *head)
{  LINKLIST  *t;      char ch;   int k=0;
   while(strings[k]!= ’$’)
    { t = (LINKLIST *) malloc(sizeof(LINKLIST));  t->data = strings[k];
      t->next = head;      head = t;      k++;
    }
    return(head);
}
main()
{ LINKLIST *head = NULL;
  head = creatlink_nohead_head(head);
  head = invertlink(head);
  while(head != NULL)
   {  printf(″  %c″,head->data);    head = head->next;}
}
```

2、阅读下列算法，说明该算法完成的功能。

```
#define MAXLEN 100
Typedef  struct   node;
  { char  data;  struct  node  *lch, *rch;} BinNode, * BinTree;

Typedef  struct  {  char  ch[30];  int  len; } str;
Typedef  struct  { BinNode  *elem[30];  int  rear,front; } quetp;
int  i=0;   quetp  q;   str  s ;

void  crt_pre(Bintree *t)
 {   char c;      c=s.ch[i];     i=i+1;
     if (c==’.’)  *t=null;
     else { *t=(BinNode *)malloc(Len);     (*t)->data=c;
            crt_pre(&(*t)->lch);           crt_pre(&(*t)->rch);
          };
```

```
  }

void level(BinNode  *bt)
{ struct BinNode  *b;
   q.front=0;     q.rear=0;    if (!bt)   return;
   q.elem[q.rear]=bt;     q.rear=q.rear+1;

    while (q.front < q.rear)
     {  b=q.elemc[q.front];   q.front=q.front+1;  printf("%c  ",b->data);
        if (b->lch!=0)
          { q.elem[q.rear]=b->lch;    q.rear=q.rear+1; }
        if (b->rch!=0)
          { q.elem[q.rear]=b->rch;   q.rear=q.rear+1; }
     }
}
main()
{char c[]={'a','b','c','.','.','.','d','e','.','.','g','.','.','.','f','.','.','.','.','.','!'};
 int k=0;  BinNode  *bt;
 for(k=0,s.len=0;c[k]!='!';k++,s.len++) s.ch[k]=c[k];
  crt_pre(&bt);      level(bt);
}
```

七、程序填空（每空 2 分，共 10 分）

下列是统计树中叶子结点数的程序，请填写适当的语句，完成该功能。

```
typedef  struct  csnode
   { char data;     struct csnode *fch,*nsib;  } csnode;

int CountLeaf (csnode *T)
{  int leaf;  csnode *T1;
   if (!T)  return （1）                ;
   else {  if (!T->fch)   return 1;
         else {  T1=T->fch;   leaf= （2）          ;
               while(T1)
                    {  leaf=leaf+ （3）              ;
                       T1=（4）              ;
                    }
               return（5）              ;
```

```
            }
        }
    }
```

第七题答题处

（1）	（2）	（3）	（4）	（5）

八、编程题（每题 10 分，共 10 分）

用 C 语言并编写简单选择排序（先选出最小数字）

一、每题 2 分，共 20 分

（1） A　　（2） A　　（3） B　　（4） B　　（5） D

（6） B　　（7） D　　（8） A　　（9） D　　（10） A

二、每题 1 分，共 10 分

1. 17/7　　2. A_{71} 和 A_{17}　　3. 4　　4. 40　　5. n(n-1)

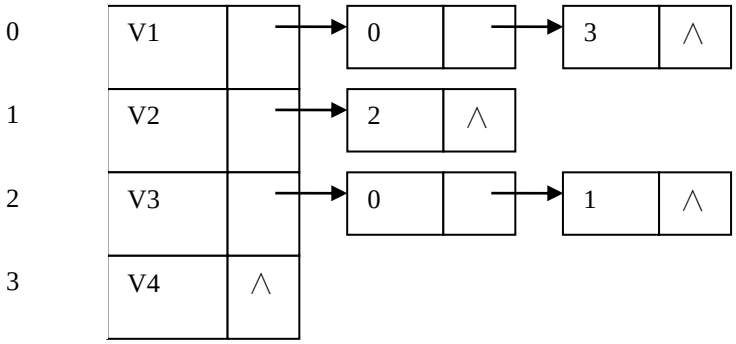
6. 9　　7. n+1　　8. 0(a+b)　　9. n-1　　10. 2e

三、每题 1 分，共 10 分

1 X；2：X；3. X；4 √；5: X；6 X；7 X；8 X；9 X；10 X

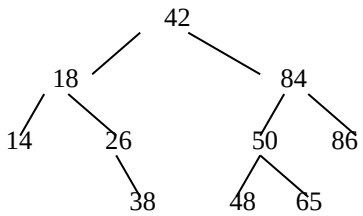
四、每题 5 分，共 20 分

1、

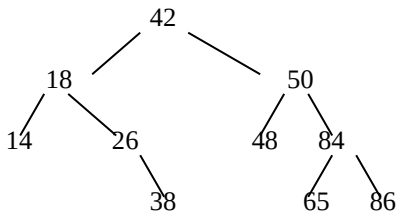


2、

1〉二叉排序树为：



2〉



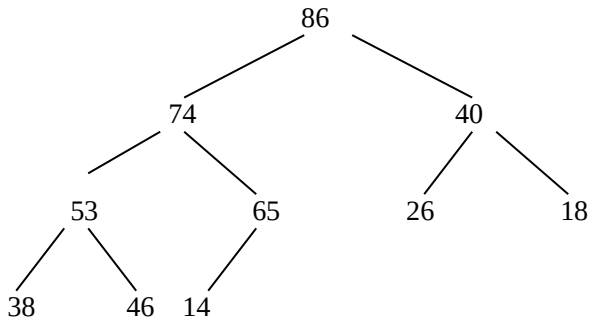
3、

散列表如下：

0	1	2	3	4	5	6	7	8	9	10	11	12

51	26	80	16	54	41			8	22		11	12

4、



五、每题 5 分，共 10 分

1、证明： 三叉树结点总数为 n= n_0+n_3 ，分支总数为 $3n_3$，分支总数+1= 结点总数

所以 $n_0+n_3=3n_3+1$

$n_3=(n_0-1)/2$

2、 叶子结点数为 $n-\left\lfloor \frac{n}{2} \right\rfloor$ ；结点总数为 n；分支结点数为 $\left\lfloor \frac{n}{2} \right\rfloor$

六、每题 5 分，共 10 分

1、建立一个不带头结点的单链表，然后再逆置它，输出结果 datastr 。

2、对二叉树进行层次遍历, 输出 a b c d e f g

七、每题 2 分，共 10 分

(1) 0; (2) 0; (3) CountLeaf(T1); (4) T1->nsib; (5) leaf;

八、每题 10 分，共 10 分

```
#include "stdio.h"
main()
 { int i,j,temp;
  static int a[10]={12,4,89,5,47,56,78,6,4,12};/*待排数列*/

  for(i=0;i<10;i++) printf("%d ",a[i]);       /* 打印待排序列*/

  for(i=0;i<10;i++)
   {k=i;
    for(j=i+1;j<10;j++)
      if (a[j]<a[k])   k=j;
    if (k!=i)  { temp=a[k]; a[k]=a[j]; a[j]=temp; }
     }
   printf("\n");
   for(i=0;i<10;i++)  printf("%d ",a[i]);   /* 打印排序完成的序列*/
   printf("\n");
 }
```

一、 单项选择(每题 2 分，共 20 分)

1、 深度为 5 的完全二叉树拥有的最少结点数为（ ）。

A. 15 B. 32 C. 16 D. 64

2、 赫夫曼树中叶子结点数为 n，则内部结点数为（ ）。

A. n B. 2n-1 C. n-1 D. n-2

3、程序段 FOR (i=n-1;i>=1;i--)

FOR(j=1 ;j<= i;j++)

IF (A[j]>A[j+1]) A[j]与 A[j+1]对换；

其中 n 为正整数，则该程序段在最坏情况下的时间复杂度是（ ）。

A. O（n） B. O(nlogn) C. $O(n^3)$ D. $O(n^2)$

4、设有一个 n×n 的对称矩阵 A，将其上三角部分按行存放在一个一维数组 B 中，第一个元素 A[0][0]存放于 B[0]中，那么对角元素 A[i][i]存放于 B 中( )处。

A. (i+3)*i / 2 B. (i+1)*i / 2

C. (2n—i+1)*i / 2 D. n(2n—i 一 1)*i / 2

5、给定 K 值，按照开放定址法处理冲突，在哈希表中进行查找，则查找不成功的标志是（ ）。

A. 求得的哈希地址上没有记录

B. 求得的哈希地址上关键字的值与 K 不同

C. 求得的哈希地址上关键字的值为零

D. 哈希表是空表

6、广义表(b,(a,（( )）, b),d,e,((i,( ),j),k))的长度和深度分别是( )。

A.5 和 4 B. 4 和 5 C. 4 和 3 D. 4 和 4

7、堆排序在最好情况下的时间复杂度为（ ）。

A. O(logn) B. O(n) C. O(n*logn) D. $O(n^2)$

8、如果 T’是由有序树 T 转换的二叉树，则 T 中结点的后根排列是 T’结点的（ ）。

A. 先序排列 B.中序排列 C. 后序排列 D. 层次排列

9、一个队列的入队序列是 ACBD,则出队序列是（ ）。

A.CBDA B.BDAC C.DACB D.ACBD

10、静态查找表与动态查找表之间的根本差别在于（ ）。

A. 它们的逻辑结构不一样

B.施加于它们之上的操作不一样

C.数据元素的类型不一样

D.存储实现不一样

二、 填空(每题 1 分，共 10 分)

1、抽象数据类型 ADT 由三部分组成：数据对象 、_____ 、基本操作。

2、模式串为 aababababaabaab,它的 next 函数值为_____ 。

3、若二叉树先序遍历的扩展序列为 BA**CD**EF***,其中*代表空链域，则二叉树的中序遍历序列为_____。

4、5 阶 B_树中的每个结点最多有___个关键字。

5、一个无向完全图中有 n 个顶点，共有_____条边。

6、含有 n 个叶子结点的满二叉树的深度为_____ 。

7、若一个算法中的语句频度之和为 $T(n)=1000n+500n\log n+n^2$，则算法的时间复杂度为_____ 。

8、假定一组记录的关键字为(46,79,56,38,40,80)，以 46 为枢轴，对其进行一趟快速排序后得到的关键字为_____。

9、在有序表（12，24，36，48，60，72，84）中折半查找关键字 72 时所需进行的关键字比较次数为_____。

10、31 个关键字进行 2-路归并排序，要得到一个有序序列，一共要进行_____趟排序。

单项选择答案：

1	2	3	4	5	6	7	8	9	10

填空答案：

1、_____ 2、_____ 3、_____

4、_____ 5、_____ 6、_____

7、_____ 8、_____ 9、_____

10、_____

三、 判断是非(每题 1 分，共 10 分)

1、有 n 个结点的不同形态的二叉树有 n!棵。（ ）

2、简单选择排序是一种不稳定的排序方法 。（ ）

3、在只有度为 0 和度为 k 的结点的 k 叉树中，设度为 0 的结点有 n_0 个，度为 k 的结点有 n_k 个，则有 $n_0=n_k+1$。（ ）

4、折半搜索只适用于有序表，包括有序顺序表和有序链表。（ ）

5、数据元素是数据的最小逻辑单位。( )

6、在一棵二叉树中，假定每个结点只有左子树，没有右子树，对它分别进行中序遍历和后序遍历，则具有相同的结果。( )

7、对于 AOE 网络，加速任一关键活动都能使整个工程提前完成。( )

8、在遍历中序线索二叉树时，非终端节点如果有左孩子，那么它的前驱是其左子树中最左下的节点。( )

9、二叉树的平衡因子只能为-1、0 或 1。（ ）

10、有向图顶点的度是指它的邻接点的个数。（　　）

判断是非答案：

1	2	3	4	5	6	7	8	9	10

四、程序填空（每空 2 分，共 10 分）

下列是图的深度优先搜索算法。请将程序中空白处填上适当的语句完成功能。

```
#include<stdio.h>
#include "malloc.h"
#define n 4
static char ch[n]={'a','b','c','d'};
static int a[n][n]={0,1,1,0,  1,0,1,1,   1,1,0,1,   0,1,1,0};
typedef   struct node
    { int adjvex;      struct node *next;  } EdgeNode;   /*邻接表中的边结点*/
typedef struct vnode
    { char  vertex;   EdgeNode *firstedge; } VertexNode[n];   /*邻接表中的顶点结点*/
int path[n],visited[n],sum=0;    VertexNode G;

void createALGraph()
/*根据邻接矩阵，建无向网的邻接表表示（序号大的邻接点排在前面）*/
{int i,j,k;   EdgeNode  *s;
 for(i=0; i<n; i++)   { G[i].vertex=ch[i];   G[i].firstedge=0;  }
  for(i=0; i<n; i++)
  { for(j=0; j<n; j++)
    { if (a[i][j]==1)
       { s= _____(1)_____;
         s->adjvex=j;_____(2)_____;  _____(3)_____;
       }
    }
  }
}

void DFS(int i)/*以 vi 为出发点，采用邻接表的 DFS 算法*/
   { EdgeNode  * p;
     printf("   %c",G[i].vertex);      visited[i]=1;_____(4)_____;
     while(p)
           { if(visited[p->adjvex]==0)  _____(5)_____;
       p= p->next;
     }
   }
void DFSTraverse()  /*DFS 算法遍历图 G，G 采用邻接表表示*/
   { int  k;
```

```
    for(k=0;k<n;k++)  visited[k]=0;
    for(k=0;k<n;k++)
            if(visited[k]==0) DFS(k);
}

main()
{  createALGraph();   DFSTraverse(); }
```

答案: (1) _____

(2) _____

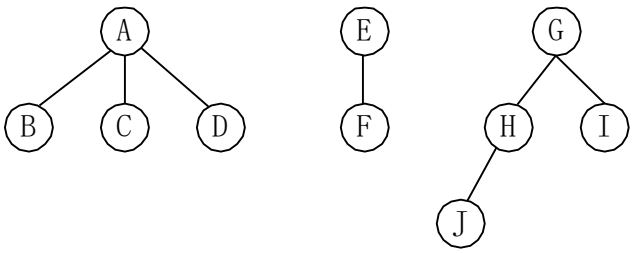
(3) _____

(4) _____

(5) _____

五、作图/解答（每题 5 分，共 20 分）

1、 树采用孩子兄弟表示法，二叉树采用二叉链表表示，二叉树和树以及森林可以相互转换，请把下图表示的森林转换成二叉树。

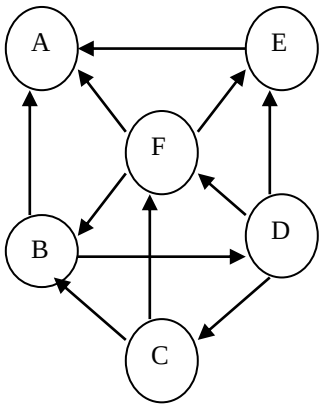


答案:

2、画出如下有向图的逆邻接表，并写出每一个顶点的入度和出度。

答案:

顶点	入度	出度	
A			
B			
C			
D			
E			
F			



3、已知一组序列为：(46, 74, 18, 53, 14, 26, 40, 38, 86, 65)，若采用堆排序方法进行排序，请构造初始堆，要求：按关键字非递减有序排列。

答案:

4、表长度为11，存储地址为a[0] 至 a[10]， 哈希函数为H(key)=key MOD 11，使用链地址法解决冲突，请画出对于给定的如下一组关键字{12，51，8，22，26，80，11，16}所得到的哈希表，并求等概率下查找不成功时的平均查找长度。

答案:

六、证明题和问答题（每题 5 分，共 10 分）

1、一棵度为 K 的树有 n_1 个度为 1 的结点，n_2 个度为 2 的结点，… …，n_k 个度为 K 的结点，问该树中有多少个叶子结点 n_0。

答案:

2、设完全三叉树中含有 n 个结点，试推导其深度公式 h。

注：等比数列的求和公式 $S_n=a_1*(1-q^n)/(1-q)$

答案:

七．算法阅读（每题 10 分，共 10 分）

阅读下列程序，说明该算法完成的功能，并写出执行后的结果。

```
void unknown (int  a[ ] , int  n)
{      int  low=0,  high=n-1;
       int  change=1;    int  temp;
       while (low<high && change)
       {   change=0;
          for  (i=low;  i<high;  i++)
             if (a[i]>a[i+1])
                  {    temp=a [i];          a[i]=a[i+1];          a[i+1]=temp;          change=1;   }
         high--;
         for (i=high; i>low; i--)
           if (a[i] < a[i-1])
                  {  temp=a[i];        a[i]=a[i-1];        a[i-1]=temp;        change=1;     }
          low++;
       }
}

main()
{
       int a[n]={7,5,8,9,0,2,1,3,6,4};
       unknown (a,10);
       for (i=0; i<10; i++) printf (“%d  “,a[i]);
}
```

答案:

八、编程题（每题 10 分，共 10 分）

1、假设二叉排序树 T 的各个元素值均不相同，设计一个递归算法按递减次序打印各元素值。用 C 语言描述二叉排序树的结构，用文字说明算法思想，并写出算法。

答案：

北 京 交 通 大 学

2015 －2016 学年第 二 学期期末考试试题

课程名称： 数据结构（A） 出题教师： 徐薇，李向前，孙永奇

专业:_____ 班级:_____ 姓名:_____ 学号:_____

题号	一	二	三	四	五	六		总分
得分								
阅卷人								

备注：请把答案写在答题纸上。

一、 填空(每空 1 分，共 10 分)

1．广义表(a, (a, b), d, e, ((i, j), k))的深度是_____。

2．一个循环队列存于长度为 n 的一维数组 B[0,n-1]中，假定队列满时，队列中有 n-1 个元素，如果 front 指向队首元素在数组中的下标，rear 指向队尾元素在数组中下一位置的下标，则求队列中元素个数的表达式为_____ 。

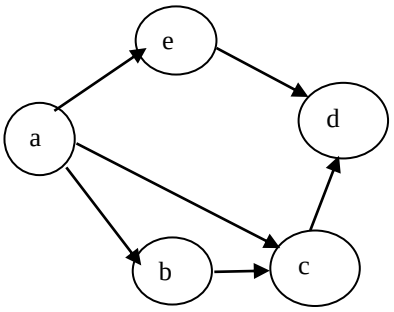
3．用S 表示入栈操作，X 表示出栈操作，若元素入栈的顺序为12345，为了得到34251的出栈顺序，相应的S和X 的操作串为_____ 。

4．一个 n 阶对称矩阵可以压缩存储在长度为_____ 的一维数组中。

5． 稀疏矩阵中的元素以三元组顺序表来表示，则三元组表中的矩阵元素（4，8，6）在相应转置矩阵中的三元组表示为_____ 。

6．若二叉树先序遍历的扩展序列为 AB*CDE***F***，其中*代表空链域，则二叉树的中序遍历序列为_____。

7．对右图进行拓扑排序，可以得到的拓扑序列共_____ 种。



8．由权值分别为 7，10，12，1，5 的叶子结点生成一棵哈夫曼树，它的带权路径长度为_____。

9．快速排序在最坏情况下的时间性能是_____。

10. 对于有 e 条边的无向图，所有顶点的度的和为_____。

二、 选择题(每题 2 分，共 20 分)

1. 下面程序段的时间复杂度是（ ）。

```
x=2;
while(x<n/2)  x=2*x;
```

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

2. 已知主串 S=‘aabaabcabaacabaab’，模式串 T=‘abaaaabab’，则 next 函数值及 nextval 函数值为（ ）。

A. 011222212 和 010222201　　B. 011222234 和 010222201

C. 011222212 和 010222104　　D. 011222234 和 010222104

3. 已知一棵含 40 个结点的二叉树中只有一个叶子结点，则该树中度为 1 的结点个数为( )。

A. 0　　B. 1　　C. 38　　D. 39

4. n 个结点的线索二叉树中线索的数目为（ ）个。

A. n-1　　B. n　　C. n+1　　D. n+2

5. 深度为 10 的二叉树拥有的最少结点数为（ ）。

A. 1023　　B. 512　　C. 511　　D. 以上都不对

6. 完全二叉树第 5 层（设根为第 1 层）有 4 个叶子结点，则完全二叉树的结点个数最多可能是 （ ）

A．19　　B.43　　C.55　　D.69

7. 对于一个具有 n 个顶点和 e 条边的无向图，若采用邻接表表示，则表头向量的大小为( ),邻接表中的全部结点总数是 ( )。

A. n+1 和 2e　　B. n 和 2e　　C. n+1 和 e　　D. n 和 e

8. 下面哪一个方法可以判断出一个有向图是否有环（ ）。

A. 求最小生成树　B. 拓扑排序　C. 求最短路径　　D. 求关键路径

9. AVL 树是一种平衡的二叉排序树，树中任一结点的( )。

A. 左、右子树的高度均相同　　B. 左、右子树高度差的绝对值不超过 1

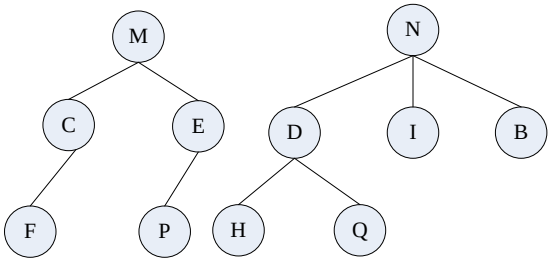
C. 左子树的高度均大于右子树的高度　D. 左子树高度均小于右子树的高度

10. 对序列{10，15，22，18，3，6，33 }用希尔排序方法排序，经一趟排序后序列变为{10，3，6，18，15，22，33}，则该趟排序采用的增量是（ ）。

A. 1　　B. 2　　C. 3　　D. 4

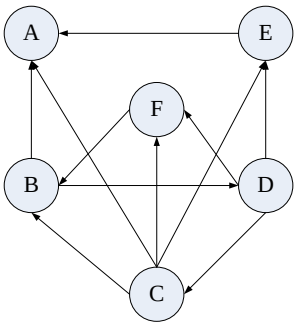
三、作图/解答(共 6 小题，共 40 分)

1. 如果树采用孩子兄弟链表表示法，树以及森林可以和二叉树相互转换，请把下图所示的森林转换成相应的二叉树。（6 分）



2. 已知一棵二叉树的层次遍历序列为 ABCDEFGHIJK 和中序遍历序列为 GKJDBEHACIF，试画出该树。（6 分）

3. 画出如下图所示有向图的逆邻接表，并写出入度最大的顶点和出度最大的顶点。（6 分）



4. 按顺序输入一组关键字序列 (70，80，90，50，40， 65，68)，画出其构成的平衡二叉树，给出过程。（6 分）

5. 设待排序的关键字序列为{12，2，16，30，28，10，16*，20，6，18}，要求排序结束后关键字非递减有序排列，画出建立初始堆的主要过程。（6 分）

6. 设有一组关键字{19,21,3,14,44,28,9,11}，采用哈希函数：H（key）=key mod 7 ，表长为 11，用开放地址法的二次探测再散列方法 Hi=(H(key)+di) mod 11(di=1^2, -1^2, 2^2, -2^2, 3^2,…,)解决冲突。要求：对该关键字序列构造哈希表，并计算查找成功时的平均查找长度。（10 分）

四、阅读下列程序，按要求回答问题。（每题 10 分，共 10 分）

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_VERTEX_NUM  50

typedef struct ArcCell {
   int adj;
} ArcCell, AdjMatrix [MAX_VERTEX_NUM][MAX_VERTEX_NUM];
typedef struct {
   int vexs[MAX_VERTEX_NUM];
   AdjMatrix arcs;
   int vexnum, arcnum;
} MGraph;
typedef struct {
   int adjvex;
   int lowcost;
} CLOSEDGE;
int total;

void init(MGraph *graph, int an, int vn)
{   int i, j;
   graph->arcnum = an;
   graph->vexnum = vn;
   for (i = 0; i < vn; i++)
   {  graph->vexs[i] = i;
      for (j = 0; j < vn; j++)
      {  graph->arcs[i][j].adj = 99999; }
   }
}
```

```
int m(CLOSEDGE *closedge, int vexnum)
{   int i, j = 99999, k;
    for (i = 0; i < vexnum; i++)
    {   if (closedge[i].lowcost != 0 && closedge[i].lowcost < j)
        {   j = closedge[i].lowcost;
            k = i; }
    }
    if (j == 99999)  return -1;
      else { total+= j; return k; }
}

void s(MGraph *graph, int u)
{   int i, j, k;
    CLOSEDGE closedge[MAX_VERTEX_NUM];
    k = u;
    for (i = 0; i < graph->vexnum; i++)
        if (i != k)
        {  closedge[i].adjvex = k;
           closedge[i].lowcost = graph->arcs[k][i].adj;
        }
    closedge[k].lowcost = 0;
    for (i = 1; i < graph->vexnum; i++)
    {   k = m(closedge, graph->vexnum);
        printf("%d -- %d\n", closedge[k].adjvex, graph->vexs[k]);
        closedge[k].lowcost = 0;
        for (j = 0; j < graph->vexnum; ++j)
            if (k != j && graph->arcs[k][j].adj < closedge[j].lowcost)
            {   closedge[j].adjvex = graph->vexs[k];
                closedge[j].lowcost = graph->arcs[k][j].adj; }
    }
}
int main()
{   MGraph graph;
    int i, arcnum, vexnum;
    int tt[11][3] = {{0, 4, 14}, {0, 1, 19}, {0, 6, 18}, {1,2,5}, {1,3,7}, {1,4,12},
{2,3,3}, {3,4,8}, {3,5,21}, {4,6,16}, {5,6,27}};
    vexnum = 7;     arcnum = 11;
    init(&graph, arcnum, vexnum);
    for (i = 0; i < arcnum; i++)
    {  graph.arcs[tt[i][0]][tt[i][1]].adj = graph.arcs[tt[i][1]][tt[i][0]].adj = tt[i][2]; }
    total=0;
    s(&graph, 0);
    printf("total=%d\n", total);
    return 0;
}
```

1. 画出这个带权网；（2 分）

2. 写出该程序完成的功能；（1 分）

3. 给出程序的输出结果。（7 分）

五、下面为二叉排序树插入结点的算法，请填空把算法补齐。（每空 2 分，共 10 分）

```
#include <stdio.h>
#include <stdlib.h>
#define  TRUE  1
#define  FALSE  0
typedef struct BiTNode {
   int data;
   struct BiTNode  *lchild, *rchild;
} BiTNode, *BiTree;

BiTree root, p;

int SearchBST (BiTree T, int key, BiTree f)
{
    if (!T)
          {  p = f;  return FALSE; }
     else  if ( key == T->data )
          {  p = T;  return TRUE; }
       else  if ( key < T->data )
                  _____________________________;   (1)
          else
                  _____________________________;   (2)
}

int InsertBST(BiTree T, int e )
{  if (!SearchBST(T, e, NULL))
  {       BiTree s = (BiTree)malloc(sizeof (BiTNode));
          _____________;_____________;   (3)
          if(!p)  { T = s;  root = T; }
             else  if ( e < p->data)
                        _____________________;  (4)
                   else
                        _____________________;   (5)
          return TRUE;
   }
  else return FALSE;
}
```

```
void main()
{       int i;    BiTree T;
        int a[] = {40, 55, 49, 73, 12, 27, 98, 81, 64, 36};
        root = NULL;
        for(i=0; i<10; i++)
        {   T = root;        InsertBST(T, a[i]); }
}
```

六、算法设计（共 10 分）

试以单链表为存储结构实现冒泡排序算法（8 分），并说明利用这种存储结构来实现该算法的优点。（2 分）

已知链表结点定义如下：

```
typedef struct Node{
   int data;
   struct Node * next;
}Node,*PNode;
```

北 京 交 通 大 学

2015 －2016 学年第 二 学期期末考试试题标准答案

课程名称： 数据结构（A） 出题教师：徐薇，李向前，孙永奇

一、填空题（每空 1 分,共 10 分）

1. 3　　2. (rear-front+n) mod n　　3. SSSXSXXSXX　　4. n(n+1)/2

5. （8，4，6）　　6. BEDCFA　　7. 3　　8. 76　　9. $O(n^2)$　　10. 2e

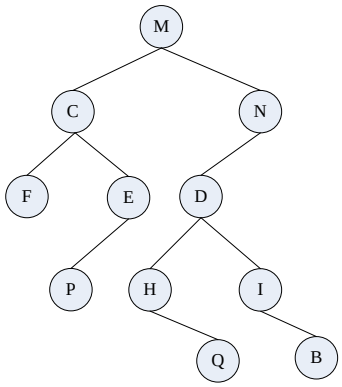
判分标准：有微小错误可给 0.5 分。

二、选择题(每题 2 分，共 20 分)

ADDCD　　CBBBC

三、 作图/解答(共 7 小题，共 40 分)

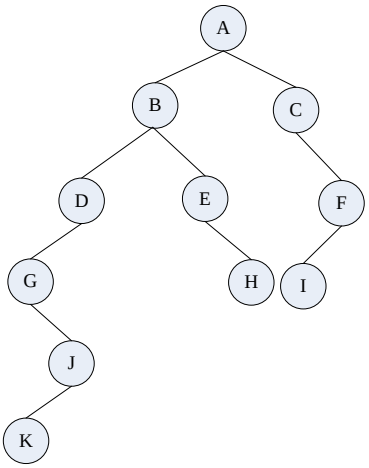
1、（6 分）



1.

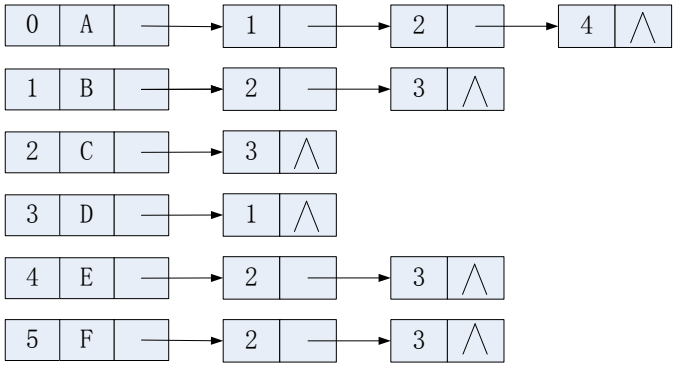
判分标准：错一个结点扣 1 分，6 分扣完为止。

2、（6 分）



判分标准：错一个结点扣 1 分，6 分扣完为止。

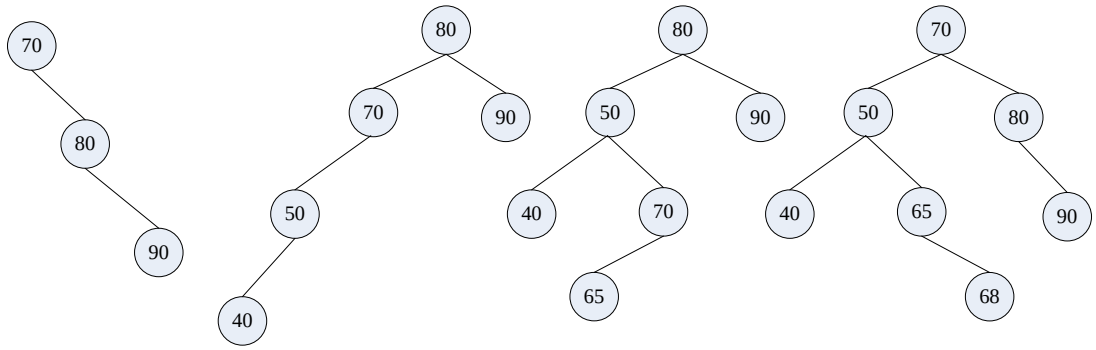
3. （6 分）



入度最大的顶点是 A，（1 分）出度最大的顶点是 C 。(1 分)

判分标准：逆邻接表 4 分：其中表头数组 1 分，表结点错 2 行扣 1 分；入度和出度最大顶点各 1 分。

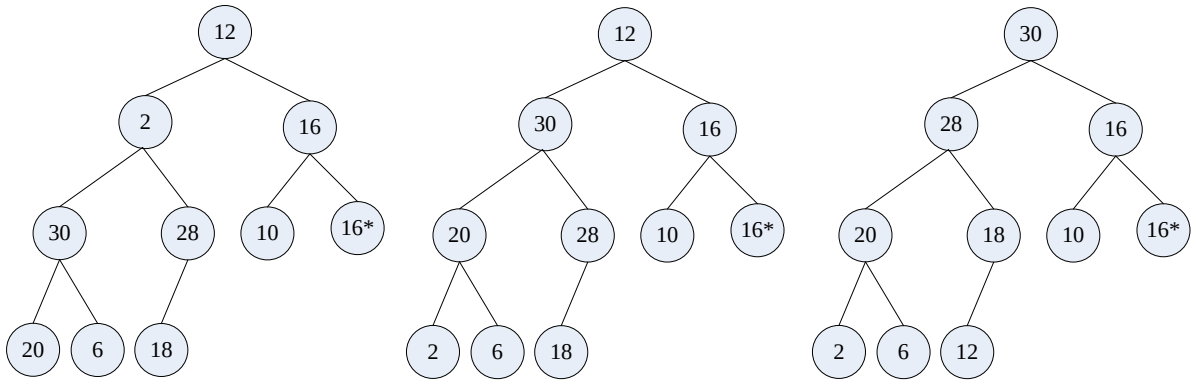
4. （6 分）



判分标准：全对给 6 分（对 2 个给 1 分），3 次平衡调整各 2 分。

5. （6 分）需要建立大顶堆排序

判分标准：全对给 6 分，3 个图每对 1 个给 2 分，若建小顶堆正确可给 2 分。



6. （10 分）

0	1	2	3	4	5	6	7	8	9	10
21	14	44	3	11	19	9				3

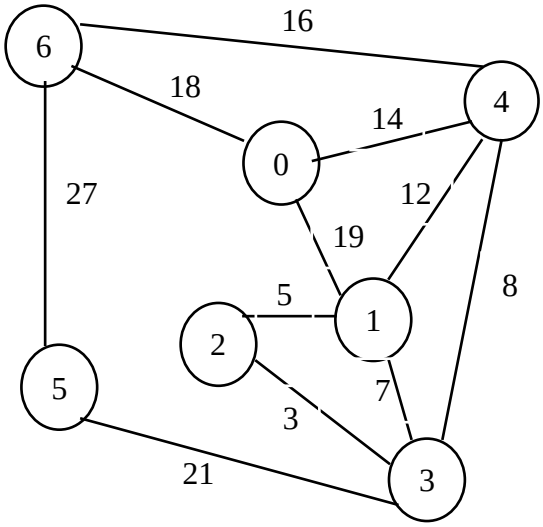
ASL=（1+2+1+1+1+1+4+3）/8=7/4

判分标准：哈希表 8 分：其中 1 个位置错误扣 1 分；ASL2 分，分子和分母对 1 个给 1 分。

四、 阅读下列算法，并回答问题：（每题 10 分，共 10 分）

判分标准：

1. 画出这个带权网；（2 分）



2. 程序完成功能:输出最小生成树（1 分）

3. 程序的输出结果:（7 分，每个 1 分）

0 -- 4
4 -- 3
3 -- 2
2 -- 1
4 -- 6
3 -- 5
total=67

五、程序填空（每空 2 分，计 10 分）

SearchBST (T->lchild, key, T); (1)
SearchBST (T->rchild, key, T); (2)
s->data = e; s->lchild = s->rchild = NULL; (3)
p->lchild = s; (4)
p->rchild = s; (5)

判分标准：完全正确，每空 2 分；若填空思路正确，但描述形式有误，每空可给 1 分。

六、算法设计题(每题 10 分，共 10 分)

1）

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Node{
    int data;
    struct Node * next;
} Node,*PNode;

PNode create_list(int n, int a[10]) //根据数组创建带头结点 head 的单链表
{    int i=0;
     PNode head, p, temp;
     head = (PNode)malloc(sizeof(Node));
     head->next = NULL;    temp = head;
     for(; i<n; i++)
    {    p = (PNode)malloc(sizeof(Node));
         p->next = NULL;    p->data=a[i];
         temp->next = p;     temp = p;
     }
     return head;
}
void BubbleSort(struct Node *head)
{    struct Node *f, *p, *x, *y;
     f = NULL;
     if(head -> next == NULL || head -> next -> next == NULL) return;
     while(f != head->next->next)
     {    for(p = head; p -> next -> next != f; p = p -> next)
          {
               if(p -> next -> data > p -> next -> next ->data)
               {
                    x = p -> next;
                    y = p -> next -> next;
                    p -> next = y;
                    x -> next = y -> next;
                    y -> next = x;
               }
          }
          f = p -> next;
     }
}
```

```
void main(){
      PNode head;
     int a[]={40, 55, 49, 73, 12, 27, 98, 81, 64, 36};
      head = create_list(10, a);
     BubbleSort(head);
}
```

判分标准：有注释（1 分）；

存储结构（1 分）；

创建链表（2 分）；

冒泡排序（4 分）：其中指针赋值（2 分）循环（1 分）边界处理（1 分）。

2）可以减少元素的移动次数。（2 分）