习题选讲

数据结构与算法分析

数据结构复习

- 复习原则：
- 1 理解各章基本概念
- 2 存储结构：
  - （1）掌握基本存储结构：线性表、栈、队列、二叉树（顺序结构、链式结构（动静态）存储信息、含义及C 语言描述） 和图（邻接矩阵和邻接表）。
  - （2）理解广义表、树和其它第（1）项中结构的其它存储结构。
- 3 算法
  - 1） 掌握基本算法（构造、销毁、插入、删除、遍历、查找、排序等）：主要步骤（或基本思想）、主要操作的实现（C语言描述），并能应用到对应问题中，能推广到相似问题中；
  - 2）复杂算法：掌握方法。理解图和树上的应用：例如最小生成树、最短路径等。
  - 3）会计算基本（简单）算法的时间复杂度和空间复杂度。

一．单项选择题

- 1、下面程序段的时间复杂度为（）
  - A. O (n)　　B. O (n^2)
  - C. O (n(n-1)/2)　　D. O (logn)

```
void fun1(int n)
{
    x = 0;
    for ( i=0; i<n; i++ )
        for ( j=i+1; j<n; j++ )
            x++;
}
```

b

- 2、若某线性表最常用的操作是删除最后一个结点，则采用存储结构算法的时间效率最高的是（　）
  - a. 单链表
  - b. 给出表尾指针的单循环链表
  - c. 双向链表
  - d. 给出表尾指针双向循环链表

d

- 3、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．s->next=p; p->next=s;

b

- 4、若一个栈以向量 S[1..n] 存储，初始栈顶指针top为n+1，则下面x进栈的正确操作是（　）
- A．top += 1; S[top]=x;
- B．S[top]=x; top += 1;
- C．top -= 1; S[top]=x;
- D．S[top]=x; top --;

C

- 5、下列说法正确的是（ ）
- (1) 稀疏矩阵压缩存储后，必会失去随机存取功能。
- (2) 若一个广义表的表头为空表，则此广义表亦为空表。
- (3) 广义表的取表尾运算，其结果通常是个表，但有时也可是个单元素值。
- (4) 从逻辑结构上看，n维数组是由多个n-1维的数组构成。
- A．仅(1)(2)　　B．仅(1)(4)
- C．仅(2)(3)　　D．仅(3)(4)

b

- 6、下列说法错误的是（ ） c
  ① 在图G的最小生成树G1中，可能会有某条边的权值超过未选边的权值。
  ② 不同求最小生成树的方法得到的生成树是相同的.
  ③ 当改变网上某一关键路径上任一关键活动后，必将产生不同的关键路径
  ④ 网络的最小生成树是唯一的
  - A12　B23　C234　D14

- 7、有n个叶子的哈夫曼树中分支节点总数为（ ）。 a
- A．n-1　　B．2n+1
- C．n+1　　D．不确定
- 解:
- n+n2 = 2n2+1
- n = n2+1
- n2 = n-1

- 8、已知一算术表达式的
- 中缀形式为：A+B*C-D/E，
- 后缀形式为：ABC*+DE/-，则其前缀形式为（ ）
  - A．-A+B*C/DE
  - B．-A+B*CD/E
  - C．-+*ABC/DE
  - D．-+A*BC/DE

d

- 9、设森林F中有三棵树，第1、第2和第3棵树的结点个数分别为M1、M2和M3（M1，M2，M3皆大于0）。与森林F对应的二叉树根结点的右子树上的结点个数是（ ）

  d

  - A．M1
  - B．M1+M2
  - C．M3
  - D．M2+M3

- 10、已知有向图G=(V,E)，其中V={ V1, V2, V3, V4, V5, V6, V7 }，E={ <V1,V2>, <V1,V3>, <V1,V4>, <V2,V5>, <V3,V5>, <V3,V6>, <V4,V6>, <V5,V7>, <V6,V7> }，G的拓扑序列是（ ）

a

- A．V1, V3, V4, V6, V2, V5, V7
- B．V1, V3, V2, V6, V4, V5, V7
- C．V1, V3, V4, V5, V2, V6, V7
- D．V1, V2, V5, V3, V4, V6, V7

- 11、下面关于折半查找的叙述中，正确的是（ ）
d
- A．表必须有序，表可以顺序方式存储，也可以链表方式存储
- B．表必须有序且表中数据必须是整型，实型或字符型
- C．表必须有序，而且只能从小到大排列
- D．表必须有序，且表只能以顺序方式存储

- 13、有n个叶子的哈夫曼树的结点总数为（　）。

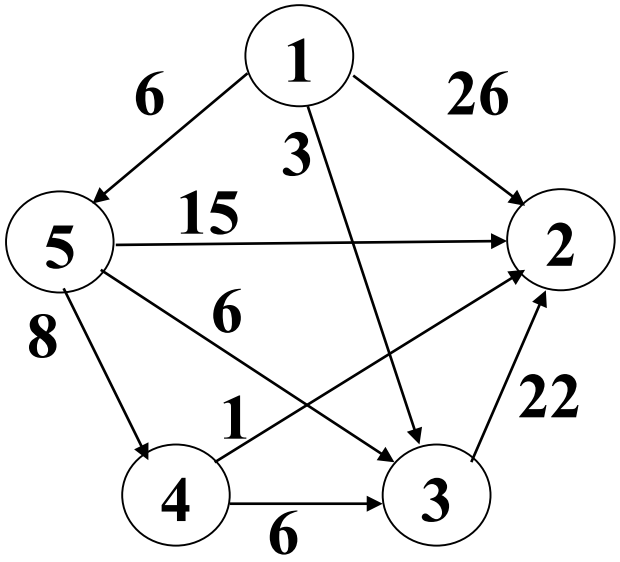
d

- A．不确定　　B．2n　　C．2n+1　　D．2n-1

- 14、下列选项给出的是从二叉树根结点分别到达两个叶结点路径上的权值序列，能够属于同一棵哈夫曼树的是（　）。
- A．36，14，5和36，14，7
- B．36，14，9和36，22，15
- C．36，14，14和36，14，9
- D．36，14，5和36，22，10

D

- 15、使用Dijkstra算法求下图中的从顶点1到其余各顶点的最短路径，将当前找到的从顶点1到顶点2、3、4、5的最短路径长度保存在数组dist中，求出第二条路径后，dist中的内容更新为（　）。
- A．26, 3, 14, 6　B．25, 3, 14, 6
- C．21, 3, 14, 6　D．15, 3, 14, 6



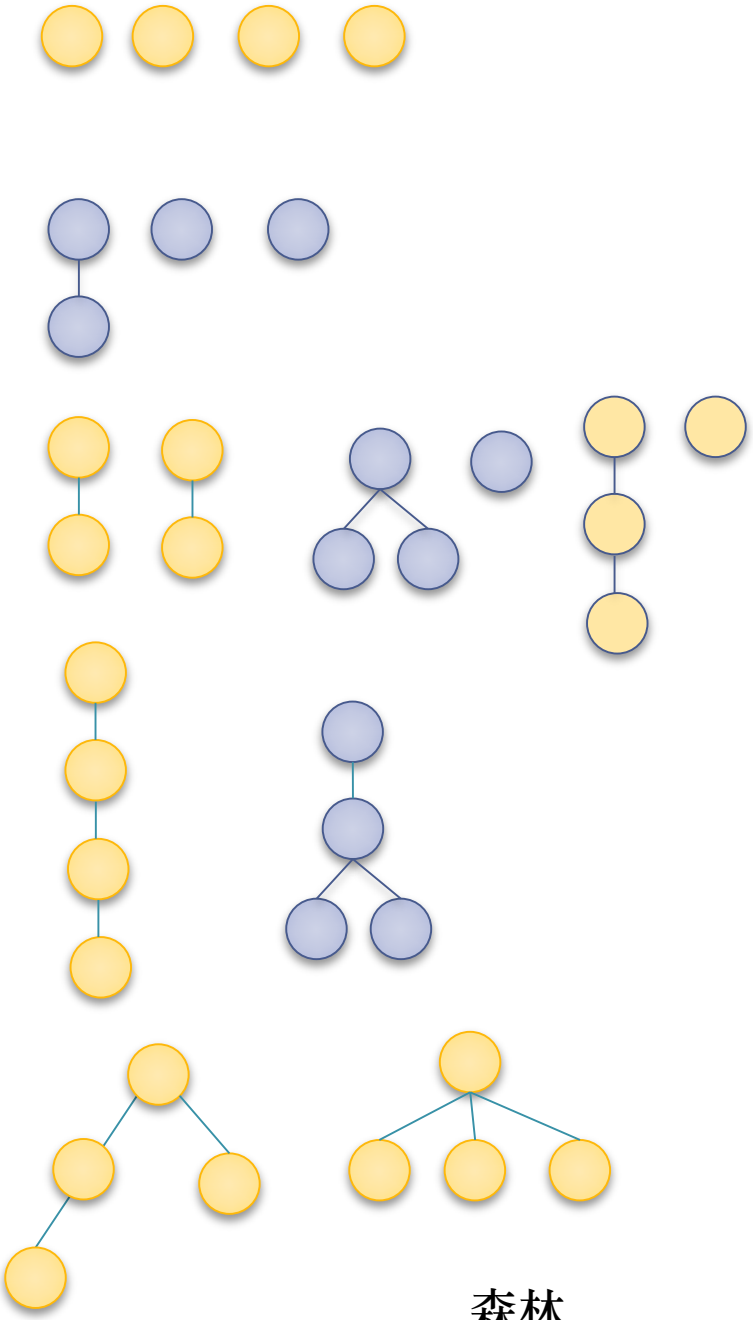
填空题

- 1、用S表示入栈操作，X表示出栈操作，若元素入栈的顺序为1234，为了得到1342出栈顺序，相应的S和X的操作串为（　）。
- 2、若用一个大小为6的数组来实现循环队列，且当前rear和front的值分别为0和3，当从队列中删除一个元素，再加入两个元素后，rear和front的值分别为（　）。

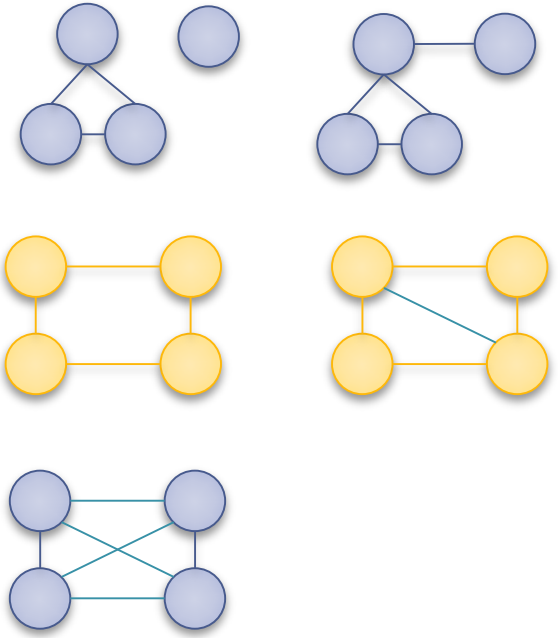
S×SS×S××　　2和4

- 4、在一棵高度为4的完全3叉树中，结点总数在（　）之间。
- 5、假设一个森林由4个节点构成，则森林可能的形态有（　）种（假设森林中的树不计顺序，树中的各个子树也不计顺序）。

14--40　　9

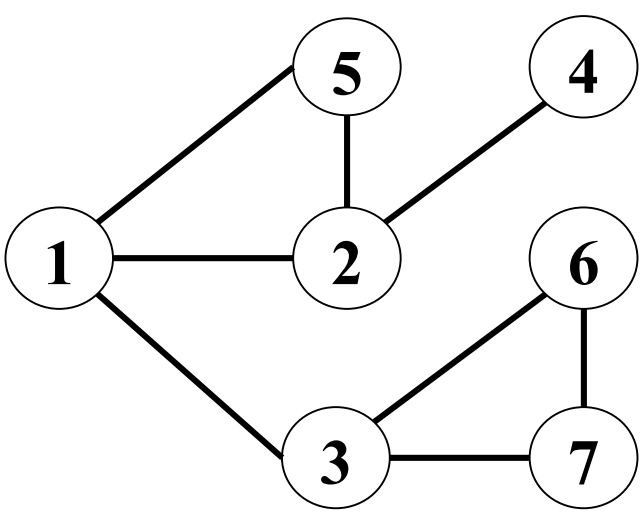


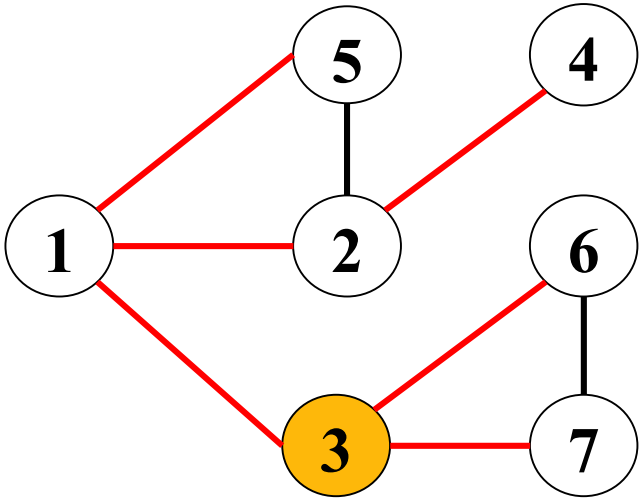
森林



无向图

- 6、无向图G(V, E)，其中V(G)={ 1, 2, 3, 4, 5, 6, 7 }，E(G)={ (1, 2)，(1, 3)，(2, 4)，(2, 5)，(3, 6)，(3, 7)，(6, 7)，(1, 5) }，对该图从顶点3开始进行遍历，去掉遍历中未走过的边，得到生成树G’(V, E)，V(G’)=V(G)，E(G’)={(1, 3)，(3, 6)，(3, 7)，(1, 2)，(1, 5)，(2, 4) }，则采用的遍历方法是（　　）。





广度优先遍历

- 7、有向图G=(V,E)，其中V(G)={ 0, 1, 2, 3, 4, 5 }，用三元组<a, b, d>表示弧<a, b>及弧上的权d。E(G)为{ <0, 5, 100>, <0, 2, 10>, <1, 2, 5>, <0, 4, 30>, <4, 5, 60>, <3, 5, 10>, <2, 3, 50>, <4, 3, 20> }，则从源点0到顶点3的最短路径长度是（　　）。

50

- 8、散列表的地址区间为0～17，散列函数为H(K) = K mod 17。采用线性探测法处理冲突，并将关键字序列：26，25，72，38，8，18，59依次存储到散列表中。元素59存放在散列表中的地址是（　）。

key	26	25	72	38	8	18	59
mod17	9	8	4	4	8	1	8

pos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
key		18			72	38			25	26	8	59					

11

- 9、向一个长度为n的向量的第i个元素(1≤i≤n+1)之前插入一个元素时，需向后移动（　　）个元素。
- 10、设有关键码序列（Q，H，C，Y，Q，A，M，S，R，D，F，X），要按照关键码值递增的次序进行排序。若采用初始步长为7的Shell排序法，则一趟扫描的结果是（　　）。

1	2	3	4	5	6	7	8	9	10	11	12
Q	H	C	Y	Q	A	M	S	R	D	F	X
Q	H	C	F	Q	A	M	S	R	D	Y	X

n-i+1

三、问答题

$$A=\begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0n-1} \\ a_{10} & a_{11} & \cdots & a_{1n-1} \\ \cdots & \cdots & \ddots & \\ a_{n-10} & a_{n-11} & \cdots & a_{n-1n-1} \end{pmatrix}$$

- 1、设有$n\times n$对称矩阵A，$a_{ij}=a_{ji}$如图所示。为了节约存储，只存对角线及对角线以上的元素。将上三角矩阵中的元素按列存放到一个一维数组B中。问：
- (1) 存放对称矩阵A上三角部分的一维数组B要有多少个元素?
- (2) 若在一维数组B中从0号位置开始存放，则矩阵A中的任一元素a_{ij}在只存上三角部分的情形下，应存于一维数组的什么下标位置？给出计算公式。

- 2、假设关键字集合为（30，15，10，40，35，43，25，28），所有关键字顺序输入。要求：构造一棵平衡二叉树排序树T（给出每插入一个节点后平衡二叉树的状态）。

- 3、设有12个初始归并段，其长度分别为8, 6, 30, 35, 62, 18, 84, 68, 11, 60, 3, 20；
- 试画出表示归并过程的最佳4路归并树，并计算树的WPL。

- 提示：先计算是否需要添加虚段，再按照大小排序
- 0，3，6，8，11，18，20，30，35，60，62，68，84

- 4、已知一棵二叉树的先序遍历序列是B, A, C, D, G, F, E和整数序列3, 2, 0, 1, 3, 0, 0。其中整数序列中的第i个数表示先序序列第i个结点的状态，含义如下：
- 0——本结点为叶结点
- 1——本结点只有左孩子
- 2——本结点只有右孩子
- 3——本结点有两个孩子
- （1）请画出该二叉树；
- （2）请说明，这样的方式能否唯一地表示一棵二叉树。若“能”，请用数学归纳法给出证明；若“不能”，请找出反例。

- 归纳法（构造性证明）
- 假设树的节点数为n（n>=0）
- 当n=0时，空树
- 当n=1时，只有根节点的树
- 归纳假设:当结点数小于k（k>1）时，结论正确。
- 下面证明n=k情况。
- 有先序列可以确定树的根。
- 情况1：如果根只有一个子节点（其状态是1或2，不可能是0）其余结点全在根左子树（或右子树）上。由归纳假设，剩余部分（结点数小于k）可以唯一确定根的左（或右）子树。
- 情况2：如果根由两个子节点（状态为3），那么在先序序列中，根的右侧结点必为根的左孩子，并可根据左孩子的状态画出左子树（结点数小于k）；类似，剩余结点就是根的右子树（结点数小于k）。

四、算法题

- 1、对单链表中元素按插入方法排序的C语言描述算法如下，其中L为链表头结点指针。请填充算法中标出的空白处，完成其功能。

```
typedef struct node
{
        int data;
        struct node * next;
} linknode, * link;
```

```
typedef struct node
{
        int data;
        struct node * next;
} linknode, * link;
```

```
void InsertSort( link L )
{   link p, q, r, u;
    p = L->next;
    L->next = NULL;
    while ( (1) )
    {   r = L;
        q = L->next;
        while ( (2) && q->data <= p->data )
        {  r = q;  q = q->next;
        }
        u = p->next;
        p->next = r->next;
        r->next = p;
        p = u;
    }
}
```

(1) p != NULL　　// 当链表尚未到尾

(2) q != NULL
// 查p结点在链表中的插入位置

(3) p->next = r->next　// 将p结点链入链表中

(4) r->next = p　　// r是q的前驱，u是下个待插入结点的指针。

- 2、请写出判断一棵二叉树是否为平衡二叉树的算法（假设不要求该二叉树是排序二叉树）。已知二叉树采用二叉链表表示，其数据定义如下：

```
typedef struct BiTNode {
    int    data; //关键字
    // 左右孩子指针
    struct BiTNode  *lchild, *rchild;
} BiTNode, *BiTree;
```

```
int IsBalanced (BiTree T ) //检查二叉树T是否是平衡树。
{// 如果不平衡则返回-1，否则返回树的深度。
        if ( !T )   return 0;
        depthLeft = IsBalanced ( T->lchild );
        if (depthLeft == -1) return -1;
        depthRight= IsBalanced ( T->rchild );
        if (depthRight == -1) return -1;
        if (depthLeft – depthRight > 1 ||
                depthRight–depthLeft > 1  ) return -1;
         return 1 + (depthLeft > depthRight ?
                depthLeft : depthRight);
}
```

- 3、计数排序算法对一个待排序的表（用数组表示）进行排序，并将排序结果存放到另一个新的表中。假设，表中所有待排序的关键字互不相同，计数排序算法针对表中的每个记录，扫描待排序的表一趟，统计表中有多少个记录的关键字比该记录的关键字小，假设针对某一个记录，统计出的计数值为c，那么，这个记录在新的有序表中的合适的存放位置即为c。
- (1) 给出适用于计数排序的数据表定义；
- (2) 使用C语言编写实现计数排序的算法；
- (3) 对于有n个记录的表，关键码比较次数是多少?
- (4) 该算法的空间复杂度是多少?

```
typedef struct
{
        int key;
        datatype info
}RecType
```

```
void CountSort(RecType a[],b[],int n)
//计数排序算法，将a中记录排序放入b中
{
  for(i=0;i<n;i++) //对每一个元素a[i]
  {     cnt=0;
        for(j=0;j<n;j++) //统计关键字比a[i]小的元素个数
                if(a[j].key < a[i].key) cnt++;
        b[cnt]=a[i];
  }
}//Count_Sort
```

其它题目

- 1、链表不具备的特点是( )。
  - A．插入和删除不需要移动元素
  - B．可随机访问任一结点
  - C．不必预分配空间
  - D．所需空间与其长度成正比

B

- 2、下面程序段的时间复杂度为（_____）

```
void fun1(int n)
{
    i=1;
    while(i<=n)
        i=i*3;
}
```

logn

- 3、下面程序段的时间复杂度为（ ） D
  - A. O (n)　　B. O (n^2)
  - C. O (n(n-1)/2)　　D. O (√n)

```
void fun1(int n)
{
    s = 0;i=0;
    while( s<n )
    {       i++ ;
            s = s + i;
    }
}
```

- 4、不带头结点的单链表head为空的判定条件是(　)。
  - A．head＝＝NULL
  - B．Head->next＝＝NULL
  - C．Head->next＝＝head
  - D．head!＝NULL

A

- 另：带头结点的单链表head为空的判定条件是(　　)。

B

- 5、带头结点的循环单链表head中，head为空的判定条件是(　)。
  - A．head＝＝NULL
  - B．head—>next＝＝NULL
  - C．head—> next＝＝head
  - D．head!＝NULL

C

- 6、在长度为n的(　)上，删除第一个元素，其算法复杂度为O(n)。

A

- A．只有表头指针的不带头结点的循环单链表
- B．只有表尾指针的不带头结点的循环单链表
- C．只有表尾指针的带头结点的循环单链表
- D．只有表尾指针的带头结点的循环双链表

- 7、若线性表中有2n个元素，算法(　)在单链表上实现要比在顺序表上实现效率更高。 A
- A．删除所有值为x的元素
- B．在最后一个元素的后面插入一个新元素
- C．顺序输出前k个元素
- D．交换其中某两个元素的值

- 9、文件局部有序或文件长度较小时，最佳排序方法是（　）。
  - A. 直接插入排序　B. 冒泡排序
  - C. 简单选择排序　D. 归并排序

A

- 1. 算法的优劣与算法描述语言无关，但与所用计算机有关。（ ）
- 2. 算法可以用不同的语言描述，如果用C 语言或PASCAL语言等高级语言来描述，那么算法实际上就是程序。（ ）
- 3. 栈和队列的存储方式，既可以是顺序方式，又可以是链式方式。（ ）
- 4. 带头结点的链队出队时不会改变头指针的值，但可能改变尾指针。（ ）
- 5. 一颗树中的叶子结点数一定等于其对应的二叉树的叶子数。（ ）

F F T T F

- 6. 线性表用链表存储时，结点间存储空间可不连续的。（ ）
- 7. 哈夫曼树是带权路径长度最短的树，路径上权值较大的结点离根较近。（ ）
- 8. 完全二叉树中，若一个结点没有左子树，则必是叶子结点。（ ）
- 9. 当待排记录按关键字从大到小或从小到大基本有序时，快速排序的执行时间最省。（ ）
- 10. 排序的稳定性是指排序算法中的比较次数保持不变，且算法能够终止。（ ）

T T T F F

- 11. 在顺序表上实现分块查找，在等概率查找情况下，其平均查找长度不仅与表的元素个数有关，而且与每一块中的元素个数有关。（ ）
- 12. 内部排序就是整个排序过程完全在内存中进行的排序。（ ）
- 13. 存在这样的二叉树，对它采用任何次序的遍历，结果相同。（ ）
- 14. 有一个小堆，堆中任意结点的关键字均小于它的左、右孩子关键字。则其中具有最大值的结点一定是一个叶子结点，并可能在堆的最后两层中。（ ）
- 15.散列表的查找效率主要取决于所选择的散列函数与处理冲突的方法。（ ）

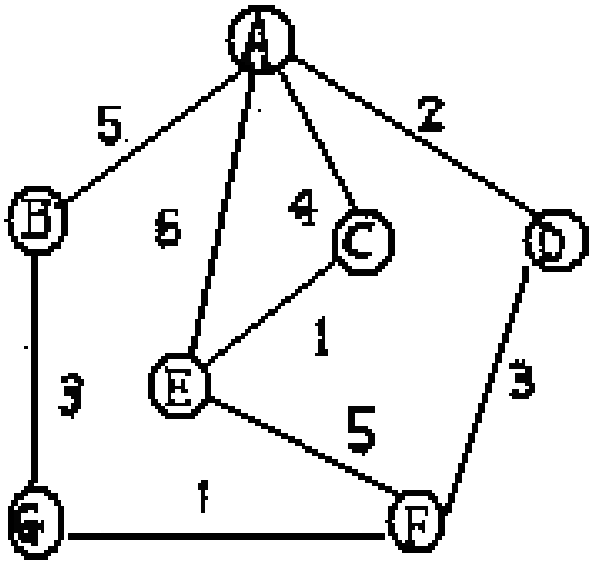
T T T T T

- 16. 顺序存储结构的主要缺点是不利于插入或删除操作。（ ）
- 17. 线性表的特点是每个元素都有一个前驱和一个后继。（ ）
- 18. 对查找进行时间分析时，只需要考虑查找成功的平均情况。（ ）
- 19. 堆是一颗完全二叉树，反之亦然。（ ）
- 20. 给定一棵树，可以找到唯一的一颗二叉树与之对应。（ ）

T F F F T

应用题

- 1、考虑下图：
- 1）从顶点A出发，求它的深度优先生成树。
- 2）从顶点E出发，求它的广度优先生成树。
- 3）根据普里姆（Prim）算法，求它的最小生成树，假定从A开始。



- 2、已知关键字输入序列{10,17,6,9,20}，画出相应的二叉排序树，并求在等概率下查找成功时的平均查找长度。

- 3、试从空树开始，画出按以下次序向3阶B-树中插入关键码的建树过程：
20,30,50,52,60,68,70.如果此后删除50和68，画出每一步执行后B-树的状态。

- 4、已知AOE网有9个结点：V1，V2，V3，V4，V5，V6，V7，V8，V9，其邻接矩阵如下：
  - (1)请画出该AOE图。
  - (2)计算完成整个计划需要的时间。
  - (3)求出该AOE网的关键路径。

∝	6	4	5	∝	∝	∝	∝	∝
∝	∝	∝	∝	1	∝	∝	∝	∝
∝	∝	∝	∝	1	∝	∝	∝	∝
∝	∝	∝	∝	∝	2	∝	∝	∝
∝	∝	∝	∝	∝	∝	9	7	∝
∝	∝	∝	∝	∝	∝	∝	4	∝
∝	∝	∝	∝	∝	∝	∝	∝	2
∝	∝	∝	∝	∝	∝	∝	∝	4
∝	∝	∝	∝	∝	∝	∝	∝	∝

算法题

1. 将顺序表中所有负数移动到表的前端，要求移动次数小。
2. 求二叉树t中两个节点u和v的最近祖先。
3. 编写按层次顺序（同一层自左至右）遍历二叉树的算法。
4. 判断二叉树t是否是完全二叉树。
5. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计一个算法，得出三个单链表，使每个单链表只包含同类字符。

1. 将顺序表R中所有n个元素分为k个区间。假设给定区间分割点为（p1, p2, …,pk-1）。分割点这样得到：找出R中的最小值MIN和最大值MAX，则分割的区间为[MIN, p1], (p1, p2], ….(pk-1, MAX]。
2. 设计在链式结构上实现简单选择排序算法。
3. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。
4. 设二叉排序树以二叉链表为存储结构，请编写一个非递归算法，从大到小输出二叉排序树中所有其值不小于X的键值。

END