**2018级A卷答案**

课程名称： 数据结构 试卷： ( A答案 ) 考试形式： 闭卷

考试对象：计算机专业2018级考试日期： 试卷：共3 页

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 题号 | 一 | 二 | 三 | 四 | 总分 |
| 得分 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 得分 |  | **一、填空题（每空1分，共10分）** |

1. 线性表是n个元素的有限序列，当（ n=0 ）时，称作空表。

2. 顺序表的缺点是（插入、删除）操作需要移动大量的元素。

3. 只设尾指针的单循环链表，找到第一个数据元素需要O( 1 )的时间。

4. 矩阵在通常（稀疏因子）<0.05时称为稀疏矩阵。

5. 给定一组数据{6，2，7，10，3，12}以它构造一棵哈夫曼树，则带权路径长度WPL的值为( 96 )。

6.已知一无向图G=（V，E），其中V={a,b,c,d,e } E={(a,b),(a,d),(a,c),(d,c),(b,e)}现用某一种图遍历方法从顶点a开始遍历图，得到的序列为abecd，则采用的是（深度优先）遍历方法。

7. 设哈希表长为14，哈希函数是H(key)=key%11,表中已有数据的关键字为15，38，61，84共四个，现要将关键字为49的结点加到表中，用二次探测再散列法解决冲突，则放入的位置是( 9 )。

8. 在一个无向图中，所有顶点的度数之和等于所有边数的（ 2 ）倍。

9. Dijkstra最短路径算法从源点到其余各顶点的最短路径的路径长度按（递增）次序依次产生。

10. 在一棵m阶B-树中,若在某结点中插入一个新关键字而引起该结点分裂,则此结点中原有的关键字的个数是（ m-1 ）

|  |  |  |
| --- | --- | --- |
| 得分 |  | **二、选择题（每题2分，共20分）** |

1. 算法的时间复杂度取决于（ C ）

A．问题的规模 B. 待处理数据的初态 C. A和B

2. 循环队列存储在数组A[0..m]中，则入队时的操作为（ D ）。

A. rear=rear+1 B. rear=(rear+1) mod (m-1) C. rear=(rear+1) mod m D. rear=(rear+1)mod(m+1)

3. 串是一种特殊的线性表，其特殊性体现在（ B ）。

A、可以顺序存储 B、数据元素是一个字符 C、可以链接存储 D、数据元素可以是多个字符

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e的运算是( C )。

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

5. 一棵二叉树高度为h,所有结点的度或为0，或为2，则这棵二叉树最少有( B )结点。

A．2h B．2h-1 C．2h+1 D．h+1

6. 二叉树的先序遍历和中序遍历如下： 先序遍历：EFHIGJK；中序遍历: HFIEJKG 。该二叉树根的右子树的根是（ C ）

A、 E B、 F　 C、 G　 D、 H

7. n个结点的线索二叉树上含有的线索数为（ C ）

A．2n B．n－l C．n＋l D．n

8. 下列哪一种图的邻接矩阵是对称矩阵？（ B ）

A．有向图 B．无向图 C．AOV网 D．AOE网

9. 下面关于求关键路径的说法不正确的是（ C ）。

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

10. 若要求尽可能快地对序列进行稳定的排序，则应选（ B ）

A．快速排序 B．归并排序 C．冒泡排序。

|  |  |  |
| --- | --- | --- |
| 得分 |  | **三、解答题（5小题，共50分）** |

1. 利用栈可以实现二叉树中序遍历的非递归算法，请回答下面的问题（共5分）

（1）什么是栈？写出顺序栈的结构类型定义，并给出栈空的条件。(3分)

（2）写出二叉树中序遍历的非递归算法思想。（2分）

（1）只允许在一端插入和删除的顺序表。

typedef struct

{

SElemType \*base; //在栈构造之前和销毁之后,值为null

SElemType \*top; //栈顶指针

int StackSize; //当前已分配的存储空间，以元素为单位

} SqStack;

栈空的条件为base==top

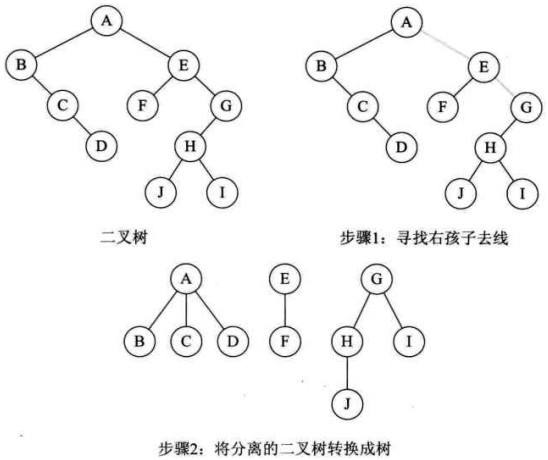
（2） 根指针进栈

当p不为空或栈非空

向左走到尽头，根节点入栈

弹出根节点访问根节点，遍历右子树

2. 回答下面问题（共10分）

（1）写出二叉树的先序遍历序列(4分)

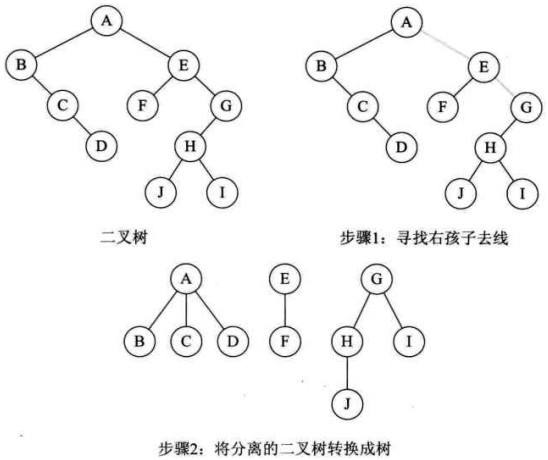
（2）转换成森林（3分）

（3）先序遍历森林（3分）

（1）ABCDEFGHJI

（2）

（3）ABCDEFGHJI

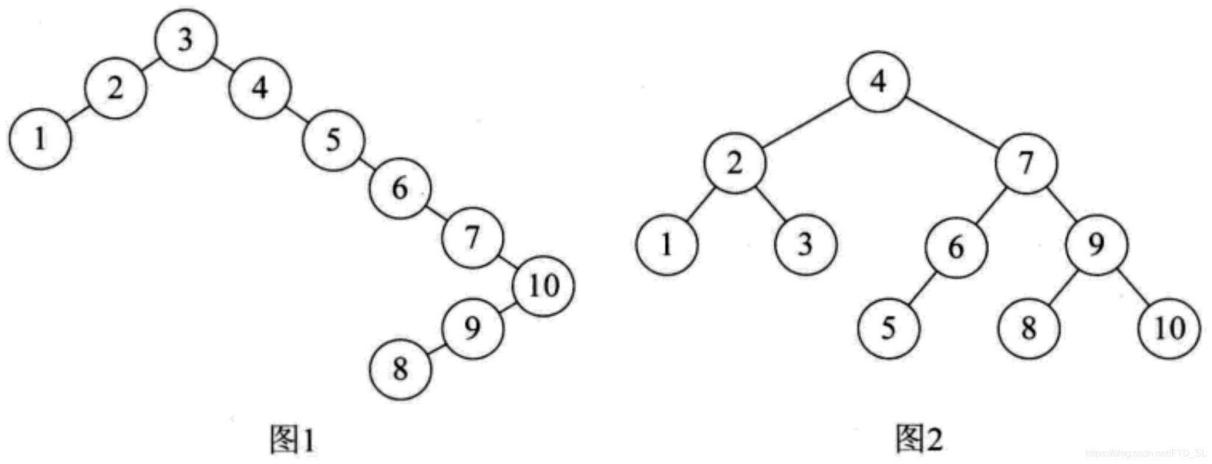


3. 现在输入序列3,2,1,4,5,6,7,10,9,8，回答下面问题（共10分）

（1）建立二叉排序树。（4分）

（2）建立平衡二叉树。（4分）

（3）分别计算查找成功时，两者的平均查找长度。（2分）



平均查找长度:二叉排序树1+4+6+4+5+6+7+8=41平衡二叉树1+4+12+12=29。

4. 有N个士兵（1<=N<=100),编号依次为1,2,...,N.队列训练时，指挥官要把士兵从低到高排成一行，但指挥官只知道“1 比2 高，7 比 5高”这样的比较结果。如图所示，回答下面问题（共10分）

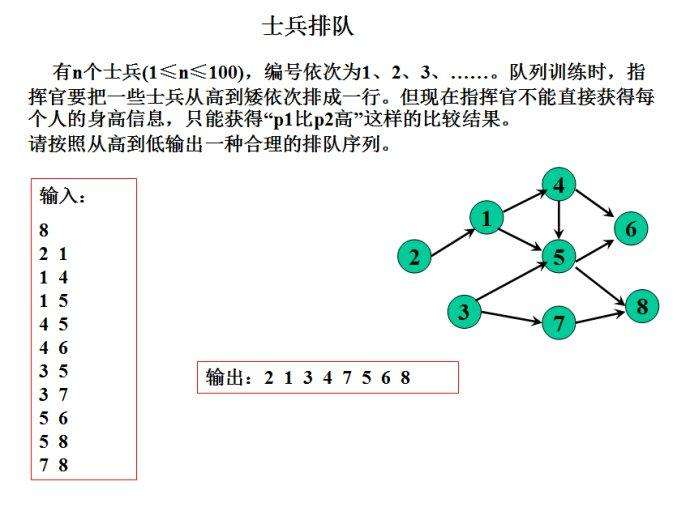
（1）给出算法思想。（3分）

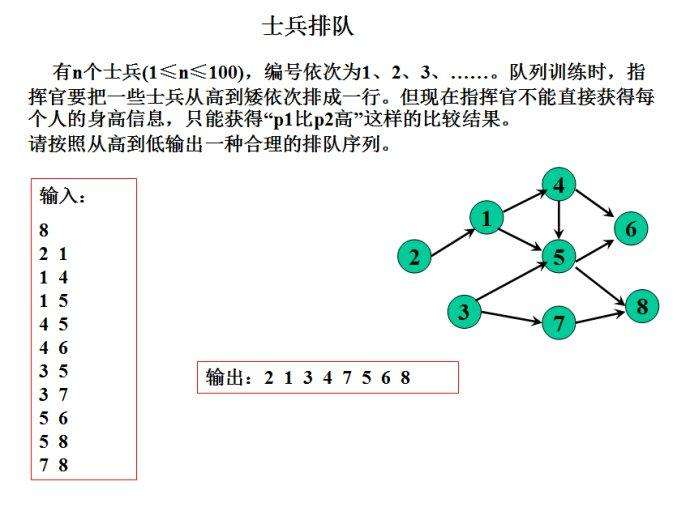
（2）求出一种合理士兵从高到低的排列。（3分）

（3）给出广度优先搜索和深度优先搜索次序。（4分）

（1）在有向图中选一个没有前驱的顶点且输出之

从图中删除该顶点和所有以它为尾的弧

重复上述两步，直至全部顶点均已输出；或者当 图中不存在无前驱的顶点为止

（2）

（3）2 1 5 4 8 6 3 7 2 1 4 5 6 8 3 7

5. 有4口矿井需要供电，在一口矿井上建立一个发电站， 费用为 v（发电站的输出功率可以供给任意多个矿井）。将这口矿井与另外的矿井之间建立电网， 费用为 p。因为不一定仅仅建立一个发电站，那么就建立一个超级源点，把所有点建立发电站的费用设为到这一点的路径长度。求出保证所有矿井电力供应的最小花费。要求画出结果，写出计算过程。（共10分）



选择在4号矿井建立发电站然后把所有矿井都与其建立电网，总花费是3+2+2+2 = 9。

6. 一组记录的关键字序列为（64，56，23，89，10，75），写出对其进行直接插入排序的过程中，每一趟排序后的结果，要求从小到大进行排序。(共5分)

起始（64），（56,23,89,10,75）

第一趟 （56,64），（23,89,10,75）

第二趟 （23,56,64），（89,10,75）

第三趟 （23,56,64,89），（10,75）

第四趟 （10,23,56,64,89），（75）

第五趟 （10,23,56,64,75,89）

|  |  |  |
| --- | --- | --- |
| 得分 |  | **四、算法题（3小题，20分）** |

1. 堆是一种常用的数据结构，回答下面问题：（共6分）

（1）什么是堆？（3分）（2）写出一个算法，判断一个长度为n的数组是不是大顶堆。（3分）

（1）堆中某个节点的值总是不大于或不小于其父节点的值；堆总是一棵完全二叉树。

（2）bool isHeap（int arr，int len）

{

for (int i = 0; i < len / 2; i++) {

if (2 \* i + 2 >= len) {

if (arr[i] >= arr[2 \* i + 1]) {

continue;

} else {

return false;

}

} else if (arr[i] >= arr[2 \* i + 1] && arr[i] >= arr[2 \* i + 2]) {

continue;

} else {

return false;

}

}

评分标准：算法基本正确但未考虑2 \* i + 2 >= len情况下得2分。

2. 已知家族谱用树的孩子兄弟法表示，求家族人口总数。（共6分）

（1）写出算法思想（3分）

通过遍历获得家族谱人口总数。

（2）写出代码实现（3分）

int count(CSTree T)

{

if(T)

{

return 1+ count (T->firstchild)+ count (T->nextsibling);

}

else

return 0;

}

评分标准：算法可采用递归算法和非递归算法，算法有语法错误得2分。

3. 为防止煤炭运入限定区域，在道路上设置了若干检查站，有检查站的道路是不允许通行的。试输出从地点i是否能运送煤炭到地点j？（共8分）

（1）写出拟采用的数据结构及其存储结构(2分)

（2）写出算法思想(3分)

（3）写出算法实现(3分)

（1）采用图表示，采用邻接矩阵存储图。

（2）顶点为路口，边为街道，有检查站的边权值设为无穷大，其他为1。从顶点出发遍历此图，权值为0的边删除（或跳过），到达j则输出true。

（3）bool DFSTraverse(MGraph G，int i,int j)

{

for(v=0;v<G.vexnum;v++)

visited[v]=FALSE; // 访问标志数组初始化(未被访问)

DFS(G,i);

if(visited[j]==true)

return true;

else

return false;

}

void DFS(MGraph G,int v)

{

visited[v]=TRUE; // 设置访问标志为TRUE(已访问)

for(w=FirstAdjVex(G,v);w>=0;w=NextAdjVex(G,v,w)))

if(G.arcs[v][w].info!=maxInt&&!visited[w])

DFS(G,w);

}