**软件学院 2021-1-16《数据结构》期末考试**

1. 有带头结点的单链表HL，向表头中插入结点p，对应的代码是（）。
2. p->next=HL.Next; HL->next=p;
3. p->next = HL; HL=p;
4. p->next.= HL; p=HL;
5. HL=p; p->next=HL;
6. 栈与队列的相同点是（）。
7. 只在端点插入、删除元素；
8. 都是先进先出的；
9. 都是先进后出的；
10. 没有任何相同点；
11. 无向图的邻接矩阵是（）。
12. 对称矩阵；
13. 零矩阵；
14. 上三角矩阵；
15. 对角矩阵；
16. 一个哈夫曼树，有个叶子，则它总共有（）个结点。

3

1

1. 对右图，哪个拓扑排序是正确的（）。
2. 125634

4

2

1. 516234
2. 123456
3. 521643

6

5

1. 对时间消耗为，它的时间复杂度是（）。
2. 深度优先搜索与二叉树的（）相似。
3. 中序遍历；
4. 先序遍历；
5. 后序遍历；
6. 层次遍历；
7. 设森林T中有三棵树，第一，第二，第三棵树的结点个数分别为N1，N2，N3。与森林T对应的二叉树根结点的右子树上的结点个数是（）。
8. N1
9. N1+N2
10. N3
11. N2+N3
12. 判断有向图中存在回路可以使用（）。
13. 广度优先搜索；
14. 最小生成树；
15. 最短路径搜索；
16. 拓扑排序；

二、解答题

1. 设栈的初始状态为空，元素、、、、、依次入栈，出栈的顺序为、、、、、。则栈的容量至少为多大？说明进栈和出栈的有关流程并说明栈中保留的元素。

2. 一棵AVL树T中结点的关键码均为正整数(从1开始取值递加，n=1, 2, 3, 4 …)，它有下列特点：

（1）左子树的高度不高于右子树；

（2）删除关键码为k1的某个叶结点，然后再插入关键码k1，得到的AVL树与原

AVL树T不同；

（3）删除T中关键码为k2的非叶结点，然后再插入关键码k2，得到的AVL树与原AVL树T相同；

（4）往T中插入某个关键码k3，然后再删除k3，得到的AVL树与原AVL树T不同。

画出具有上述特点且结点个数最少的一棵AVL树。并指出关键码k1、k2、k3的值分别是多少？

3. 已知先序遍历序列AEFBGCDHIKJ，以及中序遍历序列EFAGBCHKIJD；

（1）画出以上序列表示的二叉树；

（2）写出该树的后序遍历序列；

（3）画出该树的后序线索化树；

（4）用中文描述已知前序遍历序列和中序遍历序构造二叉树的思路；

三、编程代码题

1. 补充代码：

//下滤

void percDown(Node[] a, int i, int n){

int childe; Node tt;

for(tt=a[i]; i\*2<n; i=child){

child = 2\*i;

if(child!=n-1 && a[child]<a[child+1]){

++child;

}

if(tt<a[child]){

a[i]=a[child];

}else{

break;

}

a[i]=tt;

}

}

void HeapSort(Node[] a, int n){

for(i=\_\_\_;i>=1;i--){

\_\_\_\_\_\_\_\_\_\_\_\_;

}

for(i=n;i>1;i--){

t = a[1];

\_\_\_\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_\_\_\_;

\_\_\_\_\_\_\_\_\_\_\_;

}

2. 写出判断当前二叉树是二叉搜索树的算法；

（1）用中文详细描述思路；

（2）用C++或Java写出代码；

Class TreeNode{

int keyvalue;

int leftnode;

int rightnode;

}