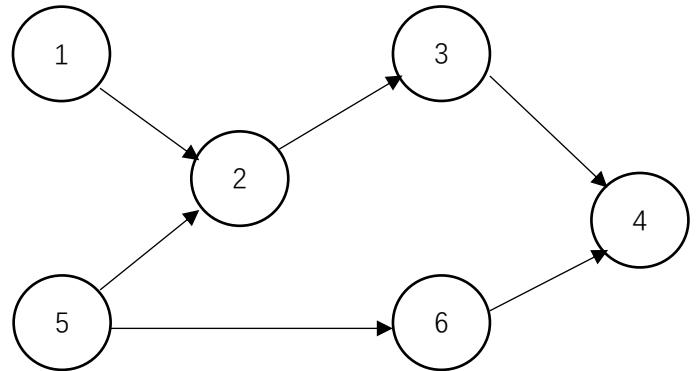


软件学院 2021-1-16 《数据结构》期末考试

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（）。
 - A. $p \rightarrow \text{next} = \text{HL}; \text{HL} \rightarrow \text{next} = p;$
 - B. $p \rightarrow \text{next} = \text{HL}; \text{HL} = p;$
 - C. $p \rightarrow \text{next} = \text{HL}; p = \text{HL};$
 - D. $\text{HL} = p; p \rightarrow \text{next} = \text{HL};$
2. 栈与队列的相同点是（）。
 - A. 只在端点插入、删除元素；
 - B. 都是先进先出的；
 - C. 都是先进后出的；
 - D. 没有任何相同点；
3. 无向图的邻接矩阵是（）。
 - A. 对称矩阵；
 - B. 零矩阵；
 - C. 上三角矩阵；
 - D. 对角矩阵；
4. 一个哈夫曼树，有 n_0 个叶子，则它总共有（）个结点。
 - A. $2n_0$
 - B. $2n_0 - 1$
 - C. $2n_0 + 1$
 - D. $3n_0$

5. 对右图，哪个拓扑排序是正确的（）。
 - A. 125634
 - B. 516234
 - C. 123456
 - D. 521643



6. 对时间消耗为 $(3n + n \log_2 n + n^2 + 8)$ ，它的时间复杂度是（）。
 - A. $O(n)$
 - B. $O(n \log n)$
 - C. $O(n^2)$
 - D. $O(\log n)$
7. 深度优先搜索与二叉树的（）相似。
 - A. 中序遍历；
 - B. 先序遍历；
 - C. 后序遍历；
 - D. 层次遍历；

8. 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（）。
- A. N1
 - B. N1+N2
 - C. N3
 - D. N2+N3
9. 判断有向图中存在回路可以使用（）。
- A. 广度优先搜索；
 - B. 最小生成树；
 - C. 最短路径搜索；
 - D. 拓扑排序；

二、解答题

1. 设栈的初始状态为空，元素 e_1 、 e_2 、 e_3 、 e_4 、 e_5 、 e_6 依次入栈，出栈的顺序为 e_2 、 e_4 、 e_3 、 e_6 、 e_5 、 e_1 。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, $n=1, 2, 3, 4 \dots$), 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 k_1 的某个叶结点, 然后再插入关键码 k_1 , 得到的 AVL 树与原 AVL 树 T 不同;
- (3) 删除 T 中关键码为 k_2 的非叶结点, 然后再插入关键码 k_2 , 得到的 AVL 树与原 AVL 树 T 相同;
- (4) 往 T 中插入某个关键码 k_3 , 然后再删除 k_3 , 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 k_1 、 k_2 、 k_3 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 child; 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;  
}
```