

南京信息工程大学期末试卷

20 23—2024学年 第 1 学期 《数据结构》 课程试卷 **A**

本试卷共 5 页；考试时间 120 分钟；出卷： 数据结构课程组 ；出卷时间 20 23 年 12 月

_____学院_____ 专业 _____ 年级 _____ 班

学号 _____ 姓名 _____ 得分 _____

一、单项选择题（每小题 2 分，共 20 分）

1. 下面程序段的时间复杂度是（ ）。

```
i=s=0;
while(s<n){
    i=1; s+=i;
}
```

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

2. 数组 $A[0..5,0..6]$ 的每个元素占 5 个字节，将其按 列优先次序 存储在起始地址为 1000 的内存单元中，则元素 $A[5][5]$ 的地址是（ ）。

- A. 1205 B. 1200 C. 1180 D. 1175

3. 针对任何一棵非空二叉树进行遍历，若限定先左后右的次序，则先序、中序和后序遍历序列中，叶子结点之间的相对次序（ ）。

- A. 发生改变 B. 保持不变 C. 不能确定 D. 以上都不对

4. 将长度为 n 的单链表链接在长度为 m 的单链表之后的算法的时间复杂度为（ ）。

- A. $O(1)$ B. $O(n)$ C. $O(m)$ D. $O(m+n)$

5. 若数据元素序列 11, 12, 13, 7, 8, 9, 23, 4, 5 是采用下列排序方法之一得到的第 2 趟排序后的结果，则该排序算法只能是（ ）。

- A. 冒泡排序 B. 插入排序 C. 选择排序 D. 二路归并排序

6. 假设栈初始为空，将表达式 $a/b+(c*d-e*f)/g$ 转换为后缀式的过程中，当扫描到 f 时，栈中的元素从栈底到栈顶依次是（ ）。

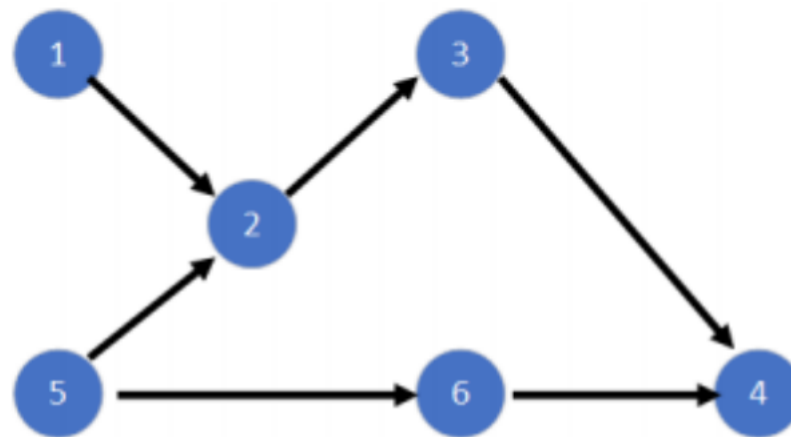
- A. $\# / + (* - *$ B. $\# / + - *$
C. $\# + (- *$ D. $\# + (* -$

7. 用顺序存储的方法，将完全二叉树中 n 个结点，按层逐个从上到下、从左到右的顺序存放在一维数组 $R[0..n-1]$ 中。若结点 $R[i]$ 有左孩子，则存放左孩子的数组元素是（ ）。

- A. $R[2i-1]$ B. $R[2i+1]$ C. $R[2i]$ D. $R[2/i]$

8. 由关键字序列 (15, 6, 17, 5, 9, 16, 4, 7, 12, 18, 8) 来构造平衡二叉树, 在构造平衡二叉树时有可能失去平衡, 则需要做 () 调整, 使得二叉排序树由不平衡转化为平衡。
- A. 先左旋后右旋 B. 先右旋后左旋 C. 单向左旋 D. 单向右旋

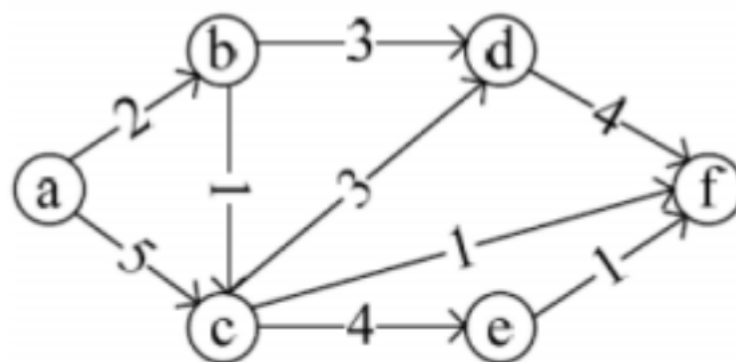
9. 针对下面的有向图进行拓扑排序, 则拓扑排序的结果序列是 ()。
- A. 125634 B. 516234 C. 123456 D. 521643



10. 若需要在 $O(n \log_2 n)$ 的时间内完成对数组的排序, 且要求排序是稳定的, 则可选的排序方法是 ()。
- A. 快速排序 B. 堆排序 C. 归并排序 D. 直接插入排序

二、填空题 (每空 2 分, 共 20 分)

- 已知一棵完全二叉树的第 6 层 (设根为第 1 层) 有 8 个叶子结点, 则该完全二叉树的结点个数最多有 (1) 个。
- 已知关键字序列为: 5, 8, 12, 19, 28, 20, 15, 22 是小顶堆, 现在插入关键字 3, 则调整后得到的小顶堆为 (2) 。
- 若将关键字 1, 2, 3, 4, 5, 6, 7 依次插入到初始为空的平衡二叉树 T 中, 则调整后 T 中平衡因子为 0 的分支结点个数是 (3) 。
- 用希尔排序方法对一个数据序列进行排序时, 若第 1 趟排序结果为: 9, 1, 4, 13, 7, 8, 20, 23, 15, 则该趟排序采用的增量 d ($1 \leq d \leq 5$) 的值可能为: (4) 。
- 针对下图所示的有向网, 若采用迪杰斯特拉 (Dijkstra) 算法, 求源点 a 到其他各个顶点的最短路径。已知得到的第一条最短路径的目标顶点为 b, 第二条最短路径的目标顶点为 c, 则按最短路径从小到大, 后续得到的其余各个最短路径的目标顶点依次为 (5) 。



- 设 F 是一个森林, B 是由 F 转换得到的二叉树, F 中有 n 个非终端结点, 则 B 中右指针域为空的结点有 (6) 个。
- 在对 n 个数据对象进行的二路归并排序中, 整个归并过程的时间复杂度为是 (7) 。

8. 设栈 S 和队列 Q 的初始状态均为空，元素 abcdefg 依次进入栈 S。若每个元素出栈后立即进入队列 Q，且 7 个元素出队列的顺序是 bdcfeag，则栈 S 的容量至少能存放 (8) 个元素。
9. 已知可以利用栈将原表达式转换为后缀式，在将表达式 $a+b-a*((c+d)/e-f)+g$ 转换为后缀式 $ab+acd+e/f-*g+$ 的过程中，同时保存在栈中的操作符（包括左括号和 #号）的最大个数是 (9) 。
10. 已知一个 6×5 的稀疏矩阵的三元组表示（按行序从小到大排列）如下图所示（i 表示行，j 表示列，e 表示非零元的值），则该矩阵的转置矩阵的三元组表示（按行序从小到大排列）是 (10) 。

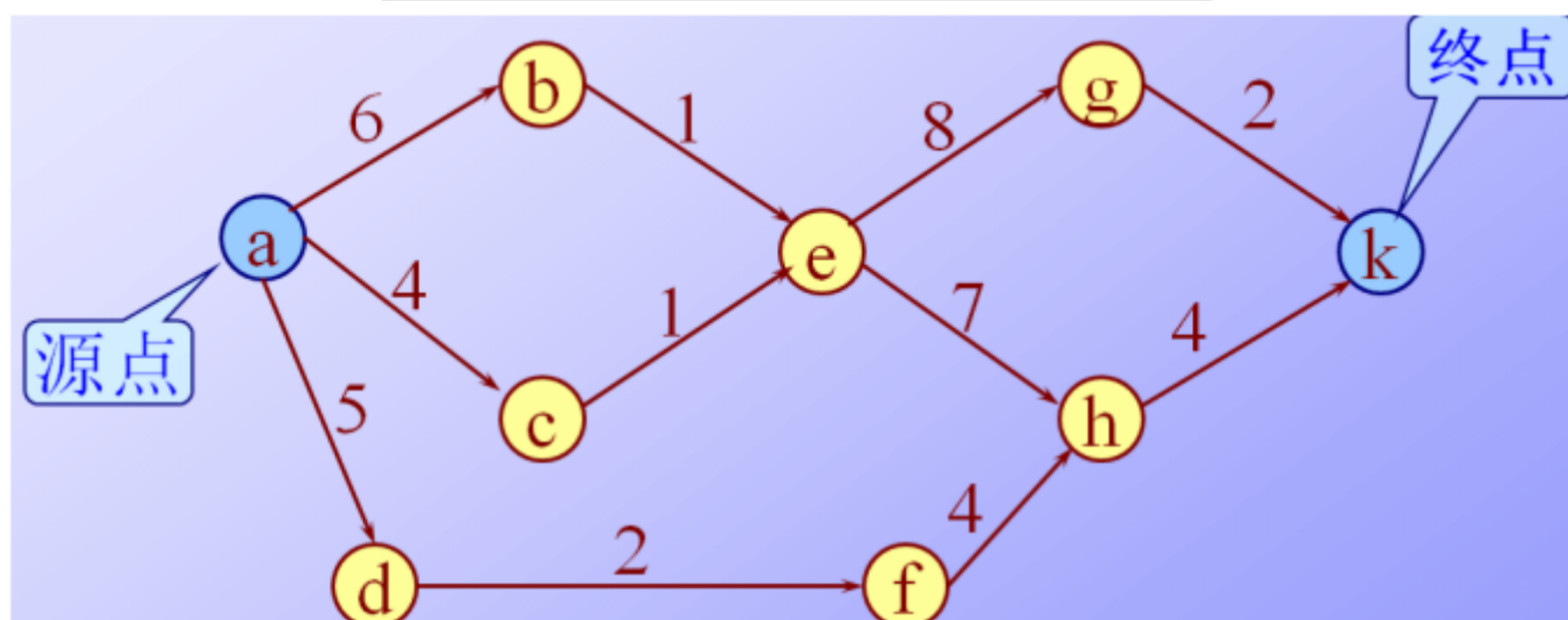
i	j	e
0	4	1
2	1	-1
3	4	-2
4	0	5
5	2	7

三、判断题（每题 2 分，共 10 分，错误打“×”正确打“√”）

1. 已知一棵二叉树的先序和后序序列，则一定能够唯一确定该二叉树的形状。 ()
2. 向一棵二叉排序树中插入一个新结点，则该结点一定为叶子结点。 ()
3. 设某堆中有 n 个结点，则在该堆中插入一个新结点的时间复杂度为 $O(\log_2 n)$ 。 ()
4. 如果某个有向图的逆邻接表中第 i 条单链表为空，则第 i 个顶点的出度为零。 ()
5. 用边表示活动的网络（AOE 网）的关键路径是指从源点到终点的路径长度最长的路径。 ()

四、应用题（本大题有 8 小题，共 40 分，每小题 5 分）

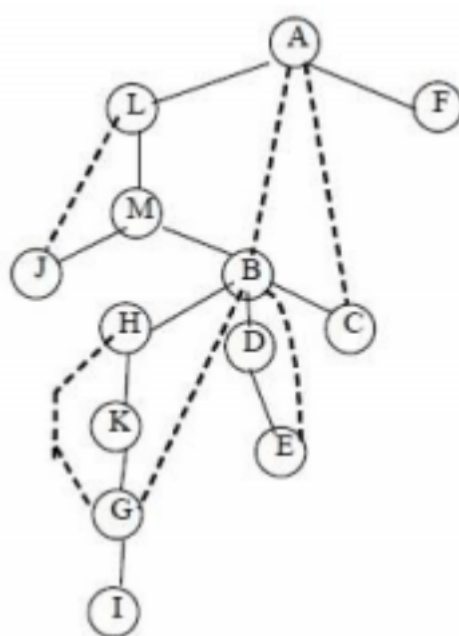
1. （5 分）下图是一个工程项目的活动图，其中顶点表示活动开始或结束，弧表示包含的活动，弧上的权值表示完成该活动所需的时间，请重画该图，并在每条弧的下方或上方标出该活动的最早发生时间和最迟发生时间（具体格式为： $A|B$ ，A 代表最早发生时间，B 代表最迟发生时间）；画出该图的所有关键路径，并计算该工程的工期。



2. (5分) 已知一个图的顶点集 V 和边集 E 分别为: $V=\{1,2,3,4,5,6,7\}$;
 $E=\{<2,1>,<3,2>,<3,6>,<4,3>,<4,5>,<5,1>,<5,7>,<6,1>,<6,2>,<6,5>\}$;
 若采用邻接表存储它, 并且每个顶点邻接表中的边结点, 都是按照顶点序号从小到大的次序链接的。请:

- (1) 试画出该图;
- (2) 试给出该图所有可能的拓朴排序的序列。

3. (5分) 已知下图为某个连通图的深度优先遍历生成树(从顶点 A 出发), 其中虚线为回边。请: 判断该连通图是否为重连通图? 如果不是, 哪些顶点是关节点?



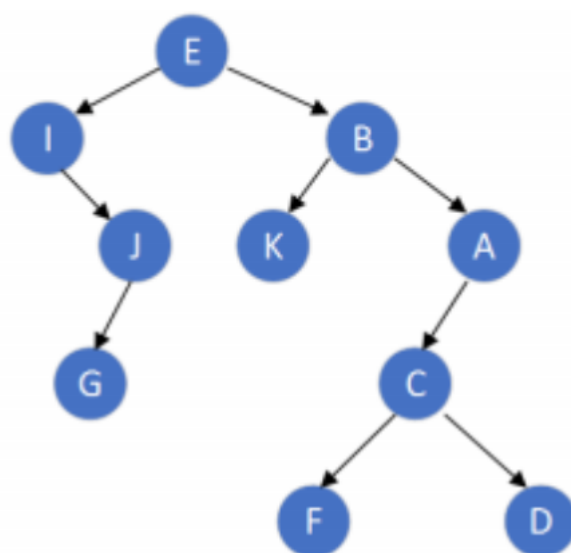
4. (5分) 假设某符号集 X 中包含 5 个符号: (s_1, s_2, s_3, s_4, s_5), 它们在报文中各自出现的频率分别为: (3, 5, 7, 9, 11)。试:

- (1) 画出该哈夫曼树(结点按照权值左小右大排列) ;
- (2) 写出各个符号的哈夫曼编码(按照左分支标“0”, 右分支标“1”的方式编码) ;
- (3) 计算该树的带权路径长度 WPL ,

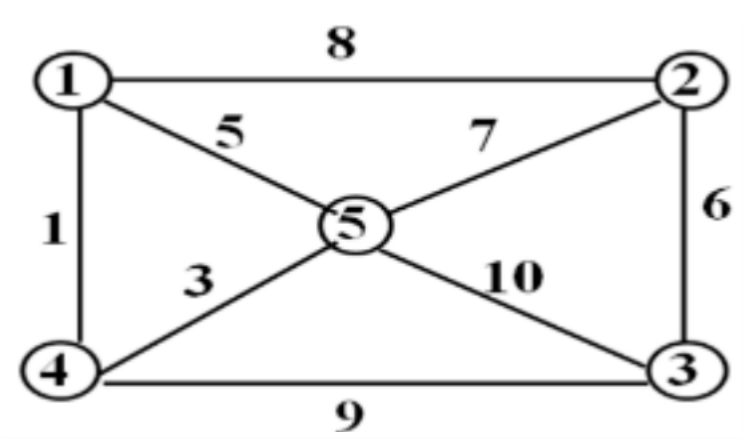
5. (5分) 已知关键字集合 $\{30, 29, 04, 14, 05, 21\}$, 设哈希函数为: $H(key) = key \text{ MOD } 7$ (表长=7), 请画出分别采用下列两种处理“哈希冲突”的方法创建的哈希表 : (1) 二次探测再散列 (又称平方探测再散列); (2) 链地址法; 并请分别计算这两个哈希表的 ASL ,

6. (5分) 下图为一棵由森林转换过来的二叉树, 请问:

- (1) 该二叉树由几棵树转换过来?
- (2) 请画出由下列二叉树转换成的森林。



7. （5 分）已知下图表示一个通讯网络， 边表示城市间的通讯线路，边上的权表示架设线路花费的代价，选择能连通每个城市且总代价最省的 $n-1$ 条线路，请使用克鲁斯卡尔方法， 画出该网的最小生成树的产生过程，并计算出总最小代价 **MinCost** 。



8. （5 分）已知一组记录的关键码为（ 49， 38， 65， 97， 76， 13， 27， 49 ），现采用快速排序的方法， 每趟都以待排序区间内的第一个记录为枢轴， 请写出第 1 趟和第 2 趟快速排序后的结果。

五、算法填空题 (共 10 分， 2 分/空)

下面为希尔排序算法的伪代码描述，请填空完成该算法。其中， dk 为每趟希尔排序所采用的增量。

```

void ShellInsert( SqList &L ,int dk ){ // 对顺序表 L 作 1 趟希尔插入排序
    for ( i=dk+1; i<=L.length; ++i ) // L.length 为顺序表中元素的个数
        if( L.elem[i] < L.elem[i-dk] ){ // L.elem 指向顺序表的基地址
            L.elem[0] = (1) ; // 暂存第 i 个元素
            for ( j=i-dk; j>0 && (2) ; j -=dk )
                (3) =L.elem[j]; // 记录后移，查找待插入位置
            L.elem[j+dk] = (4) ; //插入
        }
    } // ShellInsert

void ShellSort ( SqList &L ){ //对顺序表 L 作希尔插入排序
    int dlta[3], t=3; // t 表示希尔排序的总趟数， dlta 存放每趟希尔排序的增量
    for( i=0; i<t; i++){
        scanf( "%d", &dlta[i] );
        ShellInsert( L, (5) );
    }
} // ShellSort

```