



# 厦门大学《数据结构》课程试卷

信息学院计算机系 年级 专业  
学年学期: 主考教师: A 卷(✓) B 卷( )

## 一、(10 分) 问答题

- 1、内存中一片连续空间(不妨假设地址从 1 到  $m$ )提供给两个栈  $S_1$  和  $S_2$  使用,怎么分配这部分存储空间,使得对任一个栈,仅当这部分空间全满时才发生上溢。
- 2、在折半查找和表插入排序中,记录分别应使用哪种存储结构,并用一句话简述理由。

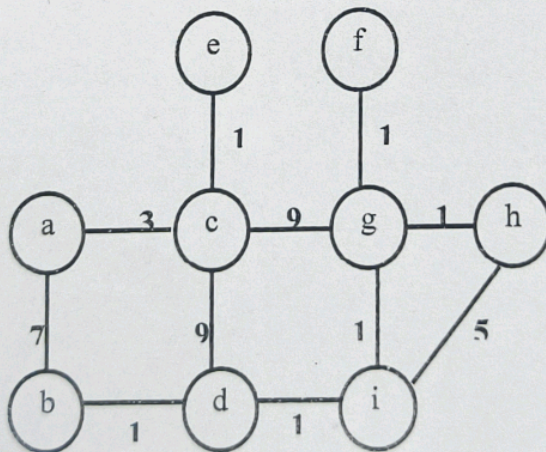
## 二、(10 分)

已知一棵二叉排序树的先序遍历序列为: 29,17,28,18,42,38,31,36,40。

- (1) 请画出该二叉排序树的二叉链表存储结构示意图;
- (2) 请用类 C 语言给出线索二叉树的存储形式描述,并画出该二叉树的带头结点的后序全线索化示意图(即后序前驱线索化和后序后继线索化)。

## 三、(15 分) 考虑下图:

- 1、请列出应用深度优先搜索 DFS 遍历,从顶点  $a$  出发得到的搜索序列。(邻接结点的访问次序以字母表顺序为准)。
- 2、请列出应用广度优先搜索 BFS 遍历,从顶点  $a$  出发得到的搜索序列。(邻接结点的访问次序以字母表顺序为准)。
- 3、当递归调用深度优先搜索遍历函数  $DFS(x)$  返回时,结点  $x$  才算终止。请依次列出各顶点终止的先后次序。(该次序和各顶点被访问的次序不同,被访问指的是调用  $DFS(x)$ ,而终止指的是  $DFS(x)$  执行结束返回。)
- 4、应用克鲁斯卡尔 Kruscal 算法添加边到最小生成树中,请按添加的先后顺序列出各条边。



## 四、(10 分)

设有一个关键码的输入序列 {55,31,11,37,46,73,63,02,07},



(1) (6分) 从空树开始构造平衡二叉树, 依次画出加入新结点时二叉树的形态。若发生不平衡, 指明平衡旋转的结果。

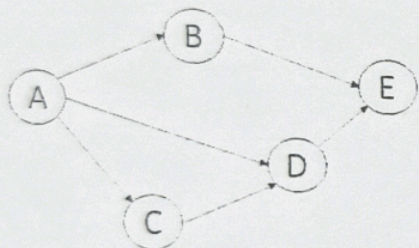
(2) (4分) 计算该平衡二叉树在等概率下的搜索成功的平均搜索长度和搜索不成功的平均搜索长度。

五、(10分) 选取哈希函数  $H(k) = (3k) \text{ MOD } 11$ 。用开放定址法处理冲突,  $di = i * (7k \text{ MOD } 10 + 1) (i=1, 2, 3, \dots)$ 。试在 0~10 的散列地址空间中对关键字序列 (22, 41, 53, 46, 30, 13, 1, 67) 造哈希表, 并求等概率情况下查找成功时的平均查找长度。

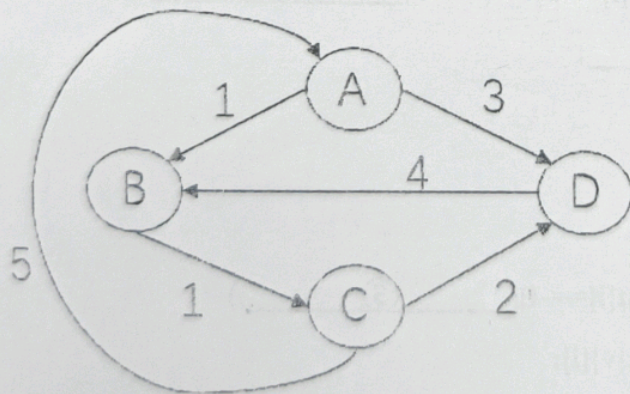
六、(15分) 设关键字序列为 {30, 90, 27, 4, 48, 15, 9, 13, 18}, 欲将其按关键字非递减排序, 请分别写出下列算法在第一趟排序结束时得到的关键字序列:

- (1) 希尔排序 (第一趟的增量为 5); (3分)
- (2) 快速排序 (以第一个关键字为枢轴); (3分)
- (3) 堆排序 (即给出堆顶为最大值的初始堆); (3分)
- (4) 二路归并排序; (3分)
- (5) 基数排序。 (3分)

七、(15分) (1) 对下图进行拓扑排序, 写出可以得到的所有拓扑序列。



(2) 用 Floyd 算法求各顶点对间的最短路径。要求: 列出各步骤的每一对顶点间最短路径长度的变化过程 (用矩阵形式表示); 以及最后求得的每一对顶点间的最短路径所经过的结点 (包括每条路径的起始点, 在 4\*4 的表格中给出)。





八、(15分) 算法题：无向连通图  $G$  有  $n$  个结点，依次编号为  $1, 2, 3, \dots, n$ 。  
用邻接矩阵的形式给出每条边的边长，要求输出以结点  $1$  为起点出发，到各结点的  
最短路径长度。请根据给定的数据结构，填写适当的 C 语言语句进行填空。

```
const int MAXV=100;

int n,i,j,v;
int w[MAXV][MAXV];
int dist[MAXV];
int used[MAXV];

int main( )
{  cin>>n;
    for (i=0;i<n;i++)
        for (j=0;j<n;j++)
            cin>>w[i][j];

    dist[0]=0;
    for (i=0;i<n;i++) dist[i]=-1;
    for (i=0;i<n;i++) used[i]=0;
    while (true)
    {
        ① _____;
        for (i=0;i<n;i++)
            if (used[i]!=1&&dist[i]!=-1&&(v==-10|| ② _____))
                ③ _____;
        if (v==-1)break;
        ④ _____;
        for (i=0;i<n;i++)
            if (w[v0][ii]!=-1&&(dist[i]==-1|| ⑤ _____))
                dist[i]=dist[v]+w[v][i];
    }
    for (i=0;i<n;i++)cout<<dist[i]<<endl;
    return 0; }
```