

杭州电子科技大学
2005 年攻读硕士研究生入学考试
<<数据结构>> 试题

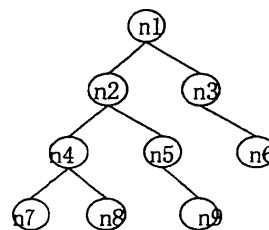
(共六大题，总分 150 分。所有答案须写在答题纸上)

一、是非题（共 20 分，每题 2 分）

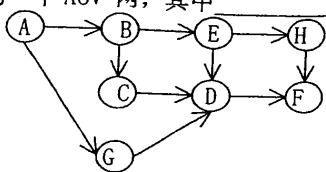
1. 完全二叉树的某结点若无左孩子，则它必是叶结点。
2. 二分查找法既适合采用顺序存储查找表，也适合链式存储查找表。
3. 算法的时间复杂性越好，可读性越差；反之，算法可读性越差，时间复杂性越好。
4. 单链表使用头节点目的是为了简化查找算法。
5. 若一个广义表的表头为空表，则此广义表亦为空表。
6. 队列是一种先进先出型线性表。
7. 邻接表可以表示有向图，也可以表示无向图。
8. 具有 1002 个结点的 Huffman 树是不存在的。
9. 二叉排序树结点中序遍历必定有序。
10. 任何一个有向图的拓扑排序的结果是唯一的。

二、选择填空（共 16 分，每空 2 分）

1. 在下列排序方法中，_____方法平均时间复杂性为 $O(n\log n)$ ，最坏情况下时间复杂性为 $O(n^2)$ ；_____方法所有情况下时间复杂性均为 $O(n\log n)$ 。
A. 插入排序 B. 气泡排序 C. 希尔排序
D. 快速排序 E. 堆排序 F. 基数排序
2. 将可利用空间表中不小于 n 且是链表中最大的空闲块的一部分分配给用户的方法为_____。
A) 最佳拟合法 B) 首次拟合法 C) 最差拟合法
3. 在需要频繁地增、删非零元时，稀疏矩阵最好采用_____存储结构。
A. 三元组表 B. 邻接表 C. 十字链表 D. 广义表
4. 把 9 个数 1, 2, 3, ..., 8, 9 填入下图所示的二叉树的 9 个结点中，使之成为二叉排序树，此时， n_2 的值是_____, n_3 的值是_____。
①. 1 ②. 2 ③. 3 ④. 4 ⑤. 5
⑥. 6 ⑦. 7 ⑧. 8 ⑨. 9
5. 下列算法中，_____适用于求图的最小生成树。
A. dfs 算法 B. bfs 算法 C. prime 算法 E. dijkstra 算法



6. 如图为一个 AOV-网，其中 为一拓朴次序。



- A. A, C, B, D, G, F, E, H B. A, B, C, G, E, H, D, F
C. A, C, D, B, E, G, F, H D. A, B, G, D, C, E, H, F

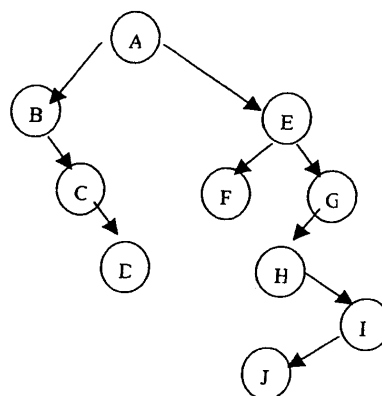
三、简答题 (共 64 分，每题 8 分)

1 右图是某森林的二叉链表 (儿子-兄弟表示法) 表示，请画出森林的图示，并给出森林的中序遍历序列。

2. 画出执行下列语句后 Q, S 内容示意图。

```

CreateQueue (Q);
CreateStack (S);
while (not end of file) {
    read number;
    if (number not 0)
        enqueue (Q, number);
    else {
        Dequeue (Q, x);
        PushStack (S, x);
    }
}
while (not IsEmptyQueue (Q)) {
    Dequeue (Q, x);
    PushStack (S, x);
}
  
```

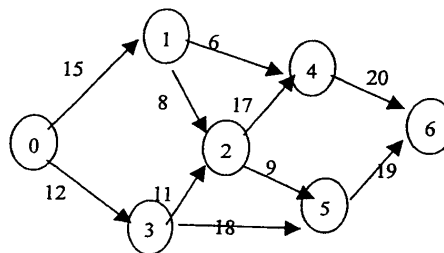


第 1 题图

输入数据为：5, 7, 12, 4, 0, 4, 6, 8, 67, 34, 23, 5, 0, 44, 33, 22, 6, 0.

3. 某二叉树有 10 个节点，它的前序序列和中序序列分别为 ABCDEFGHIJ 和 BCDAFEHJIG，画出该二叉树和它的中序线索二叉树。

4. 右图所示是某工程 AOE 网，各活动弧的单位为天，请问整个工程的工期为多少？哪些活动是关键活动？



第 4 题图

5. 设某电文只出现 a, b, c, d, e, f, g, h, i, j 共 10 个字母；出现频率分别为 12%, 15%, 10%, 05%, 04%, 13%, 5%, 7%, 9% 及 20%，请画出 Huffman 树并给出相应各字母的哈夫曼编码。

5. 哈希表的设计需要考虑哈希函数构造和冲突处理，请各举出 4 种哈希函数构造和冲突处理方法。

· 4-2 ·

7. 已知一个大小 512 字节的内存，假设依次有 4 个用户提出大小分别为 23, 52, 11 和 100 的内存分配请求，此后大小为 11 和 23 的占用块顺序被释放。假设以伙伴系统实现动态存储管理，初始所有内存空闲。请问每个用户得到的内存块起始地址是什么？画出最后的内存状态示意图。

8. 右图为一棵 3 阶 B-树，
 ① 画出该树上插入元素 22, 25 后的 B-树。
 ② 接着，再删除元素 35，画出删除后的 B-树。

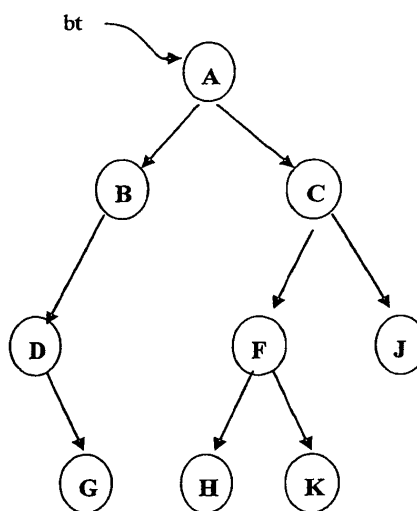
四、算法阅读：(12 分)

阅读算法 ABC，回答下列两个问题。

1. 算法的作用。(6 分)
2. 算法作用在二叉树 bt 后结果。(6 分)

```
struct Node {
    ElemType data;
    Node *lson, *rson;
};

void ABC (Node *bt)
{
    CreateStack (S); // 建栈
    if (bt != NULL) {
        PushStack (S, <bt, 0>); // 入栈
        while (!IsEmpty (S)) {
            // 出栈，指针、标记分别放入 p、tag 中
            PopStack (S, <p, tag>);
            if (tag == 0) {
                PushStack (S, <p, 1>);
                if (p->lson)
                    PushStack (S, <p->lson, 0>);
            } else {
                Print (p->data); // 打印
                if (p->rson)
                    PushStack (S, <p->rson, 0>);
            }
        }
    }
    DestroyStack (S); // 销毁栈
}
```



五、算法填空：(12 分)

填写合适内容，完成二叉排序树插入算法。

```
struct Node {
    ElemType data;
    Node *lson, *rson;
};
```

· 4 - 3 ·

```

Node *Insert (Node *bt, ElemType x) { //返回新根节点
    q = new Node; //生成新节点
    q->data = x;
    q->lson = q->rson = NULL;
    father = NULL;
    p = bt;
    while (p != NULL) {
        father = p;
        if (x < p->data)
            _____ (1) _____;
        else
            _____ (2) _____;
    }
    if (_____ (3) _____)
        return q;
    if (_____ (4) _____)
        father->lson = q;
    else
        father->rson = q;
    return bt;
}

```

六、 算法设计 (26 分)

1. 编写算法，将单链表中所有负节点（数据为负）排在其它节点之后；同时，保持非负节点间、负节点间原相对次序。（13 分）

节点类型定义如下：

```

struct Node {
    int data;
    Node *next;
};

```

2. 编写算法，判断邻接表表示的无向图是否是连通图（每一对节点均有路径相连）。（13 分）

图有关类型定义如下：

```

struct ArcNode {
    int toVertex; //到达顶点下标
    ArcNode *nextArc; //指向下一条弧信息
};
struct VertexNode {
    ElemType data; //顶点元素
    ArcNode *firstArc; //指向第一条弧信息
};
struct GRAPH {
    VertexNode vertexs [MAXSIZE]; //顶点数组
    int vertexCount; //顶点数
};

```

· 4 - 4 ·