

2010-2011 学年第二学期《数据结构》试卷 A 卷

授课班号: 242101 专业: 计算机 2009 级 学号 _____ 姓名 _____

题号	一	二	三	四	五	总分	审核
题分	15	30	12	8	35		
得分							

得分	评阅人

一、填空(15 分, 每空 1 分)

1. 数据结构是指数据及其相互之间的 联系
_____。当结点之间存在 M 对 N (M: N) 的联系
时, 称这种结构为 图。

2. 与顺序表相比, 链表的主要特点是 插入、删除 操作的效率比较高。

3. 逻辑上的线性结构我们称为线性表, 线性表即可以用顺序表存储, 也可以用链表存储。现在在某个应用程序中需要使用线性表, 在程序中会经常对线性表的元素进行随机访问, 在该应用程序中的线性表应选用 顺序 存储结构。

4. 队列的插入操作是在队列的 首部 进行, 删除操作是在队列的 尾部 进行。

5. 广义表 $A = (a, (a, b), ((a, b), c))$, 则它的深度为 4, 它的长度为 3。

6. 一个二叉树按顺序方式存储在一个一维数组中, 如图

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	D		E	F		G			H		I	J

结点 E 的左孩子节点是 H。

6. 对图的遍历主要有两种方法, 这两种方法是 深度优先 和 宽度优先。

7. 对于无向图 G，若用邻接矩阵 A 表示，则 G 的第 k 个顶点的度等于 A 的第 k 行的非零元素个数，若用邻接表表示，则 G 的第 k 个顶点的度等于 与第 k 个顶点相关联的链表的长度。
8. 为得到一棵排序二叉树的有序序列，应该对该二叉树进行 中序 遍历。
9. 对于一个关键字序列 $\{k_1, k_2, \dots, k_n\}$ ，若有 $k_i = k_j$ ($i < j$)，排序之前 k_i 在 k_j 之前，若某种排序方法使得排序后 k_j 在 k_i 之前，则称这种排序方法是 不稳定。
10. 5 阶 B_ 树中，每个结点最多有 4 个关键码，最少有 2。
11. 在线性表的散列存储中，处理冲突的常用方法有 开放定址法 和 链地址法 两种。

得分	评阅人

二、选择(30 分,将你的选择填在下表中)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

1. 下列函数在最坏情况下的时间复杂度是 [D]

```
void unknown(int n)
{
    int i,n;
    for(i=n;i>0;i--)
        for(j=i-1;j<=n;j++)
            cout<<i;
}
```

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

2. 设单链表中结点的结构为 (data, next)。链表的首指针为 first，指针 s 指向被插入节点 (用 *s 表示)，则将 *s 插入链表首部，则应执行下列 [A] 操作？

- A. $s \rightarrow next = first; first = s;$ B. $s \rightarrow next = first \rightarrow next; first \rightarrow next = s;$
 C. $s \rightarrow next = first \rightarrow next; first = s;$ D. $first \rightarrow next = s; s \rightarrow next = first;$

3. 设有单循环链表，指针 $rear$ 指向链表尾部，现在要在链表尾部插入节点 s ，则应该执行下面哪一种操作？ [**A**]

- A. $s \rightarrow next = rear \rightarrow next; rear \rightarrow next = s; rear = s;$ B. $s \rightarrow next = rear; rear \rightarrow next = s; rear = s;$
 C. $rear \rightarrow next = s; s \rightarrow next = rear \rightarrow next; rear = s;$ D. $s \rightarrow next = rear \rightarrow next; rear = s; rear \rightarrow next = s;$

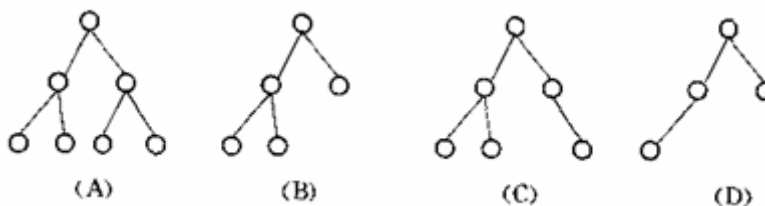
4. 设有一个顺序栈 S ，元素 s_1, s_2, s_3, s_4, s_5 依次进栈，如果 5 个元素的出栈顺序为 s_3, s_4, s_2, s_5, s_1 ，则顺序栈的容量至少应为 [**C**]

- A. 6 B. 5 C. 4 D. 2

5. 以顺序表实现的循环队列， $front$ 表示队头， $rear$ 表示队尾，队列长度为 n ，顺序表下标从 0 开始，则队列中当前元素个数是 [**C**] 计算

- A. $rear - front$ B. $front - rear$
 C. $(rear - front + n) \% n$ D. $front + 1 - rear$

7. 在下面的 4 棵二叉树中， [**C**] 不是完全二叉树



8. 已知某二叉树的后序遍历序列是 $acbed$ ，中序遍历序列是 $abcde$ ，则这棵二叉树的先序遍历序列是 [**B**]

- A. $dbeac$ B. $dbace$ C. $deabc$ D. $debca$

9. 在一棵二叉树的二叉链表表示中，假设 B_p 表示非空指针域的个数， K_p 表示空指针域个数，则 K_p 与 B_p 的关系是 []。

- A. $K_p = B_p + 1,$ B. $K_p = B_p + n_0 + 1,$ C. $K_p = B_p,$ D. $K_p = B_p + n_0 - 1$

10. 任何一棵二叉树的叶节点在先序、中序和后序遍历中，其相对次序 [**A**]

- A. 不发生改变， B. 会逆序改变 C. 会随机改变 D. 不确定

11. AOV 网是一种 (**D**)。

- A. 有向图 B. 无向图 C. 无向无环图 D. 有向无环图

- 12.在 AOE 网络中，关键路径指的是[A]
- A.从源点到汇点的最长路径 B.从源点到汇点的最短路径
- C.最长的回路 D.最短的回路
13. 随机生成 100 万个数据，使用[C]排序速度最快。
- A. 选择排序 B. 插入排序 C. 快速排序 D. 归并排序
14. AVL 树是一种平衡的二叉排序树，树中任意节点的[C]
- A. 左、右子树的高度均相同, B. 左子树的高均大于右子树的高度
- C.左、右子树的高度差的绝对值不超过 1, D. 右子树的高均大于左子树的高度
15. 对于一个具有 n 个结点和 e 条边的无向图，若采用邻接表表示，则所有边链表中边结点的总数为[B]。
- A: $e/2$ B: e C: $2e$ D: $n+e$

得分	评阅人

三、算法与程序设计 (10 分)

1. (3')下述递归程序的功能是 计算二叉树叶子节点个数

```
int xy ( BinTreeNode * T )
{
    if ( t == NULL ) return 0;
    else if ( t->left == NULL && t->right == NULL ) return 1;
    else return xy( t->left ) + xy( t->right );
}
```

2. (4')下面是用 c++语言编写的对不带头结点的单链表进行就地逆置的算法, 请完成程序。

```
void List_reverse(ListNode *L) //ListNode 表示链表节点
{
    p=L;
    _____;
```

```

while(p!=NULL) {
    s=p;
    p=p->next;
    _____;
    L=s;
}
}

```

3. (3')有序表的二分查找，返回待查找的关键字 key 的数组下标。

```

#define N 1000 (或某整数)
int Search_Bin(int A[N], int key) //数组下标为 1..N-1;若查找不成功，返回为 0
{
    int low, high, mid;
    low =1, high=N-1;
    while( _____ ) {
        mid=_____ ;
        if (key == S[mid]) return mid;
        else if ( key < S[mid]) high=mid-1;
        else _____ ;
    }
    return 0;
}

```

连通图的深度优先遍历算法 (4')

```

void DFS(Graph G, int v)
{
    // 从顶点 v 出发，深度优先搜索遍历连通图 G
    visited[v] = TRUE;
    VisitFunc(v); //对顶点 v 进行处理
}

```

```

    for(w=FirstAdjVex(G, v); [ w!=0 ]; w=NextAdjVex(G, v, w))
        if (!visited[w])
            [ DFS(G, w); ]
} // DFS

```

拓扑排序算法(4')

```

#define M 50

void TopSort(int A[M][M], int vexnum, int TopSeq[])
{
    //TopSeq[] 保存拓扑序列的数组, indegree[] 保存各顶点的入度
    int indegree[M];
    FindInDegree(G, indegree); //求各顶点的入度
    InitStack(S); //初始化一个栈 S
    for(v=0; v< vexnum; v++) { //所有入度为 0 的顶点进栈
        if([indegree[v]==0]) Push(S, v);
    }
    count=0;
    while(!StackEmpty(S)) {
        Pop(S, v); TopSeq[count]=v; count++; //输出拓扑序列
        for(k=0; k<vexnum; k++) {
            if(A[v][k]!=0) [indegree[k]--];
            if(indegree[k]==0) Push(S, k);
        }
    }
}

```

【写出算法程序（8'）（两题中任选一题）】

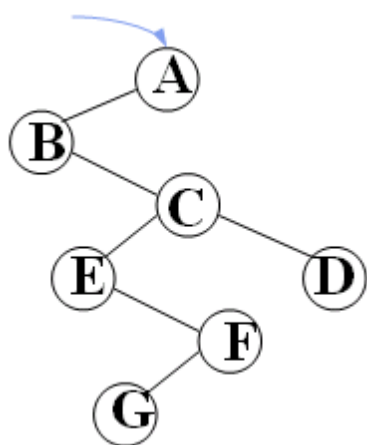
1.写出快速排序算法程序。

2.写出二叉树的非递归先序遍历算法

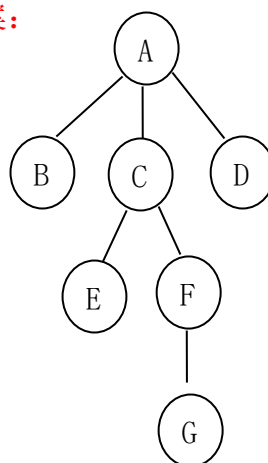
得分	评阅人

五、综合应用（36 分）

1.(6')下图是二叉树是由一查普通树转换而来，将其还原成普通树的形式



答案：



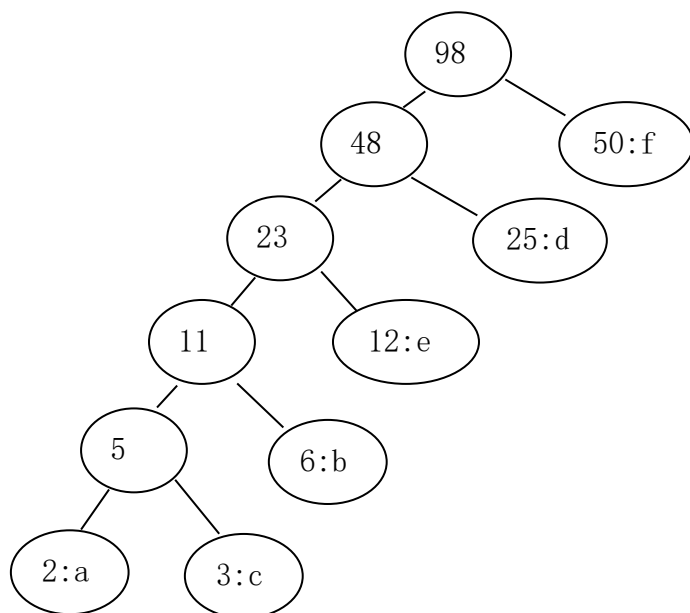
2. (8') 假设用于通信的电文仅由 5 个符 (a, b, c, d, e, f) 组成, 这 8 个字符的频率为如表所示

a	b	c	d	e	f
2	6	3	25	12	50

(1) 画出 huffman 树 (权值小的节点在左边, 权值大的节点在右边)

(2) 写出每个字符的编码 (左分支编码为 0, 右分支编码为 1)

答案



编码结果

a:00000

b:0001

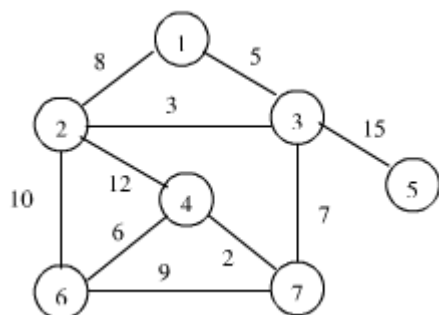
c:00001

d:01

e:001

f:1

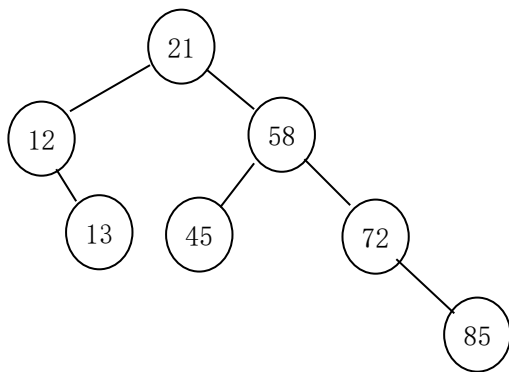
3. 对下面的无向带权图 G, 写出其最小生成树 (要求从顶点 1 出发)

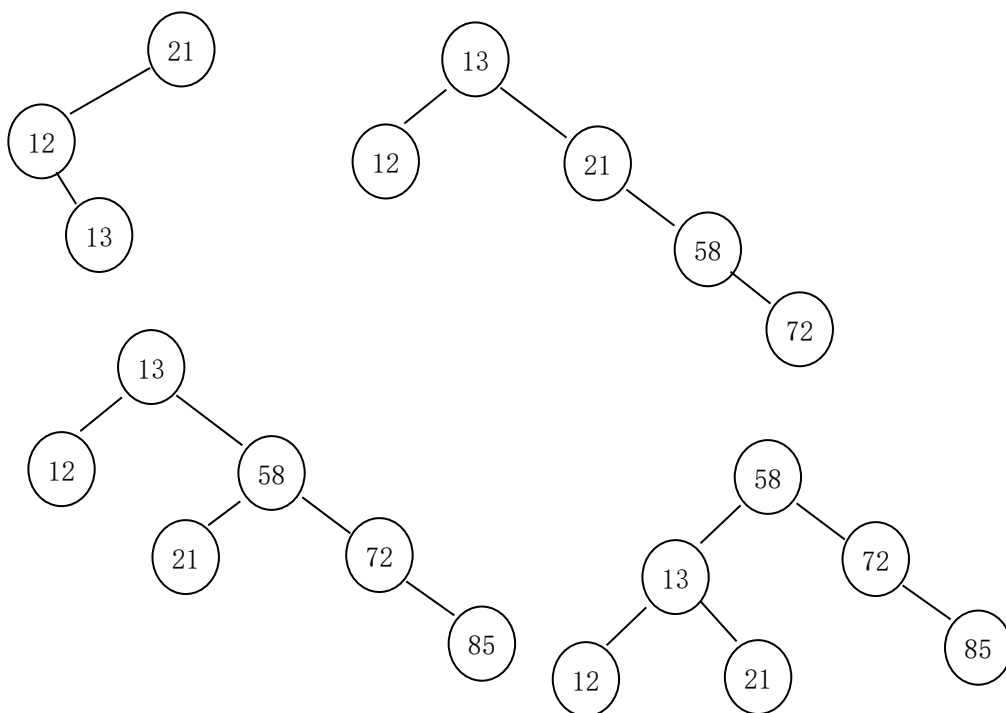


4. (6') 给定数据序列 (21, 12, 13, 58, 45, 72, 85)

(1) 构造二叉排序树

(2) 构造平衡二叉排序树 (画出旋转过程)





5. 设哈希表长度为 11，哈希函数 $h(x)=x\%11$ ，给定的关键字序列为：12, 23, 33, 45, 38, 55, 49, 28, 62.

(1) 用哈希函数计算每个关键字的地址，将关键字填入下表中（如果关键字有冲突，按顺序将其填入相同的空格中）

0	1	2	3	4	5	6	7	8	9	10
0, 55	12 , 45	23			38 , 49	28	62			

(2) 画出用线性探测法解决冲突，构造的哈希表

0	1	2	3	4	5	6	7	8	9	10

6. 给定数据序列 (42, 76, 157, 137, 93, 24, 159, 12, 121, 11)

(1) 写出第一趟快速排序的结果

第一趟结果

11, 12, 24, 42, 93, 137, 159, 157, 121, 76

(2) 构建初始大顶堆

