

2012-2013 学年第二学期《数据结构》试卷 A 卷

授课班号 242101 专业: 计算机 2011 级 学号 _____ 姓名 _____

题号	一	二	三	四	五	总分	审核
题分	15	30	30	9	16		
得分							

得分	评阅人

一、填空(15 分, 每空 1 分)

第1章 数据结构的三个要素是_____、数据存储结构和_____。

2. 与链表相比, 顺序表的主要特点是_____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存储, 也可以链式存储。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进行插入和删除操作, 那么在该应用程序中的线性表应选用_____存储结构。

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

5. 广义表 $A = (a, (a, b), ((a, b), c))$, 它的深度为_____。

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 D 的右孩子节点是_____。

6. 对图的遍历主要有两种方法, 这两种遍历方法是_____和_____。

7. 对于有向图 G, 若用邻接矩阵 A 表示, 则 G 的第 k 个顶点的出度等于_____。

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_____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_____。

10. 5 阶 B_ 树中, 每个结点最多有_____个关键码。

11. _____ 在线性表的散列存储中, 处理冲突的常用方法有_____和 _____ 两种。

得分	评阅人

二、选择题 (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[B]

```
void unknown(int n)
{
    while(n) {
        cout<<n;
        n=n/2;
    }
}
```

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 代表被插入节点, 则将 s 插入链表首部, 则应执行下列[B]操作?

A. $s \rightarrow next = first \rightarrow next; first \rightarrow next = s;$ B. $s \rightarrow next = first; first = s;$
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, 现在要在链表首部插入节点 s, 则应该执行下面哪一种操作? [A]

A. $s \rightarrow next = rear \rightarrow next; rear \rightarrow next = s;$ 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. $s \rightarrow next = rear \rightarrow next; rear = s; rear \rightarrow next = s;$

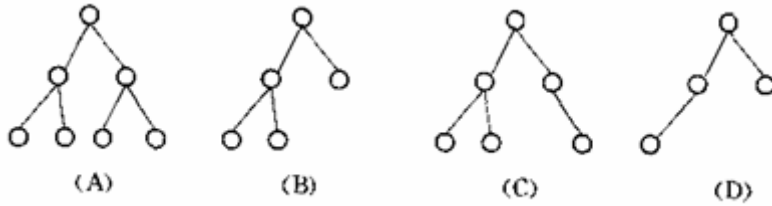
4. 设有一个顺序栈 S, 元素 a1, a2, a3, a4, a5 依次进栈, 如果 5 个元素的出栈顺序为 a4, a5, a3, a2, a1, 则顺序栈的容量至少应为[C]

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 表示列尾, 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 [] 计算

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

6. 在下面的 4 棵二叉树中, []不是完全二叉树



7. 已知一棵二叉树的按层次遍历序列是 ACEDFGM, 中序遍历序列是 DCFAEGM, 则该二叉树的前序遍历序列是[]

- A. AECDFMG B. ACDEFGM C. ACDFEGM D. ACDFEMG

8. 具有 132 个结点的完全二叉树叶子节点的个数为[C] (根的层次号为 1)

- A. 5 B. 65 C. 66 D. 127

9. 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

10. 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

11. 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

12. 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

13. 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

14. 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[]。

- A: e/2 B: e C: 2e D: n+e

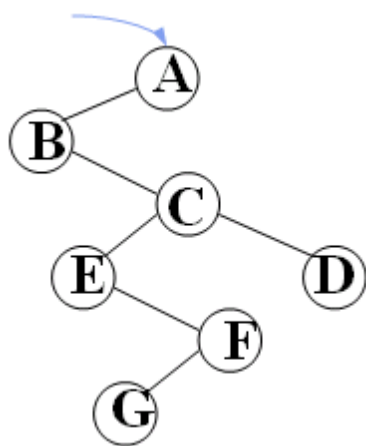
15. 图中有关路径的定义是[]。

- A. 由顶点和相邻顶点序偶构成的边所形成的序列 B. 由不同顶点所形成的序列
C. 由不同边所形成的序列 D. 上述定义都不是

得分	评阅人

三、综合应用（30 分，每题 分）

1. 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



2. 假设用于通信的电文仅由 6 个符（a, b, c, d, e, f）组成，这 6 个字符的频率为如表所示

a	b	c	d	e	f
2	6	5	30	15	60

- (1) 画出 huffman 树（权值小的节点在左边，权值大的节点在右边）
 (2) 写出每个字符的的编码(左分支编码为 0，右分支编码为 1)
 (3) 对下面的二进制串解码

3. 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)

4. 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

(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

5. 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

(2) 构建初始大顶堆

得分	评阅人

四、算法与程序设计 (25 分)

(1-3 题程序填空，每题 4 分，4,5 两题写程序，任选一题)

1. (3') 下述递归程序的功能是_____

```
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. (6') 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(_____) {
        s=p;
        p=p->next;
        _____;
        L=s;
    }
}
```

得分	评阅人

五、算法设计 (在下面 4 题中任选 2 题，每题 8 分，可以用伪代码描述，也可以用 C 语言描述)

1. 写出图的深度优先遍历算法
2. 写出简单插入排序算法
3. 写出快速排序算法程序。
4. 排序二叉树的插入算法

答案

1. 图深度优先搜索算法

```
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);
    for(w=FirstAdjVex(G, v); w!=0;w=NextAdjVex(G, v, w))
        if (!visited[w])
            DFS(G, w);
    // 对 v 的尚未访问的邻接顶点 w 递归调用 DFS
} // DFS
```

非连通图深度优先遍历

```
void DFSTraverse(Graph G)
{
    // 对图 G 作深度优先遍历。
    for (v=0; v<G.vexnum; ++v)
        visited[v] = FALSE; //访问标志数组初始化
    for (v=0; v<G.vexnum; ++v)
        if (!visited[v])
            DFS(G, v); // 对尚未访问的顶点调用 DFS
}
```

2. 插入排序

//对数组 a[]进行插入排序

```
void sort_insert(int a[],int n)
{
    int i;
    for(i=1;i<=n-1;i++)
        insert_order(a, i, a[i]);
}
```

```
void insert_order(int a[],int s,int key)
{
    int i;
    for(i=s-1;i>=0;i--) //从数组的后面向前面比较，寻找插入位置
    {
        if(key < a[i])
            a[i+1]=a[i]; //将 a[i]想后移动一个位置
        else break;
    }
```



```

    }
    a[i+1]=key;
}

```

3 快速排序

```

void QSort (Type R[], int s, int t )
{
    // 对记录序列 R[s..t]进行快速排序
    if (s < t) {
        // 长度大于 1
        pivotloc = Partition(R, s, t);
        // 对 R[s..t] 进行一次划分, 并返回枢轴位置
        QSort(R, s, pivotloc-1); // 对低子序列递归排序
        QSort(R, pivotloc+1, t); // 对高子序列递归排序
    } // if
} // Qsort.

```

```

int Partition (Type R[], int low, int high)
{
    pivotkey = R[low]; // 枢轴记录关键字
    while (low<high) { // 从表的两端交替地向中间扫描
        while(low<high && R[high]>=pivotkey) --high;
        R[low++] = R[high];
        while (low<high && R[low]<=pivotkey) low++;
        R[high--] = R[low];
    } //while
    R[low] = pivotkey; // 枢轴记录移到正确位置
    return low; // 返回枢轴位置
}

```

4. 排序二叉树插入算法

```

void BSTinsert(BSNode *T,T key)
{
    BSNode<T> *s;
    if(T==NULL) {
        T=new BSNode<T>; T->data=key;
        return ;
    }
    elseif(key <= T->data)
        BSTinsert(T->lchild, key);
}

```

```
    else  
        BSTinsert(T->rchild, key);  
}
```