

杭州电子科技大学

2015 年攻读硕士学位研究生招生考试

《数据结构》试题

(试题共 六 大题, 共 11 页, 总分 150 分)

姓名_____报考专业_____准考证号_____

【所有答案必须写在答题纸上, 做在试卷或草稿纸上无效!】

一、是非题(正确填 T, 错误填 F, 本大题共 6 小题, 每小题 1 分, 本大题共 6 分)

- (1) 从逻辑上可以把数据结构分为线性结构和非线性结构两大类()
- (2) 数据结构基本操作设置的最重要准则是, 实现应用程序与存储结构的独立。()
- (3) 线性表的特点是每个元素都有一个前驱和一个后继。()
- (4) 用树的前序遍历和中序遍历可以导出树的后序遍历。()
- (5) 一个有向图的邻接表和逆邻接表中结点的个数可能不等。()
- (6) 归并排序在任何情况下都比所有简单排序速度快。()

二、单项选择题(本大题共 23 小题, 每小题 2 分, 本大题共 46 分)

- (1) 下列程序段的时间复杂度是 ()

```
count=0
```

```
for(k=1;k<=n;k*=2)
```

```
for(j=1;j<=n;j++)
```

```
count++;
```

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

- (2) 某线性表常发生的操作为删除第一个数据元素和在最后一个元素后添加新元素, 采用 () 作为存储结构, 能使其存储效率和时间效率最高。

- A 单链表
- B 仅用头结点的循环单链表
- C 双向循环链表
- D 仅用尾指针的循环单链表

- (3) 在非空双向循环链表中 q 所指的链结点前面插入一个有 p 指的链结点

的过程是依次执行语句 $p \rightarrow rlink = q; p \rightarrow llink = q \rightarrow llink; q \rightarrow llink = p;$ ()

- A $q \rightarrow rlink \rightarrow llink = p;$
- B $q \rightarrow llink \rightarrow rlink = p;$
- C $p \rightarrow rlink \rightarrow llink = p;$
- D $p \rightarrow llink \rightarrow rlink = p;$

(4) 执行完下列语句段后，i 值为 ()

```
int f(int x){
    return ((x>0)? x*f(x-1):2)
}
```

....

int i;

i=f(f(1));

- A 2
- B 4
- C 8
- D 无限递归

(5) 对于顺序循环队列，正确的是 ()

- A 无法判断队列是否为空
- B 无法判断队列是否为满
- C 队列不可能满
- D 以上说法都不对

(6) 元素 a, b, c, d, e 依次进入初始为空的栈中，直到所有元素都出栈，则在所有可能的出栈序列中，以元素 d 开头的序列个数是 ()

- A 3
- B 4
- C 5
- D 6

(7) 顺序循环队列放在一维数组 $A[0 \dots M-1]$ 中，end1 指向队头元素，end2 指向队尾元素的后一个位置。假设队列两端均可进行入队和出队操作，队列最多能容纳 $M-1$ 个元素，初始时为空。下列判断队空和队满的条件中，正确的是 ()。

- A 队空： $end1 == end2$; 队满： $end1 == (end2+1) \bmod M$;
- B 队空： $end1 == end2$; 队满： $end2 == (end1+1) \bmod (M-1)$;
- C 队空： $end2 == (end1+1) \bmod M$; 队满： $end1 == (end2+1) \bmod M$;
- D 队空： $end1 == (end2+1) \bmod M$; 队满： $end2 == (end1+1) \bmod (M-1)$;

(8) 已知一棵有 2011 个结点的树，其叶子结点个数为 116，该树对应的二叉树中无右孩子的结点个数是 ()。

- A 115
- B 116
- C 1895
- D 1896

(9) 若 X 是后序线索二叉树中的叶结点，且 X 存在左兄弟结点 Y，则 X 的右线索指向的是 ()。

- A X 的父结点
- B 以 Y 为根的子树的最左下结点
- C X 的左兄弟结点 Y
- D 以 Y 为根的子树的最右下结点

(10) 将森林 F 转换为对应的二叉树 T, F 中叶子结点的个数等于 ()。

- A T 中叶结点的个数
- B T 中度为 1 的结点个数
- C T 中左孩子指针为空的结点个数
- D T 中右孩子指针为空的结点个数

(11) 一棵哈夫曼树共有 215 个结点, 对其进行哈夫曼编码, 共能得到 () 个不同的码字。

- A 107 B 108 C 214 D 215

(12) 若平衡二叉树的高度为 6, 且所有非叶节点的平衡因子均为 1, 则该平衡二叉树的结点总数为 ()。

- A 10 B 20 C 32 D 33

(13) 若用邻接矩阵存储有向图, 矩阵中主对角线以下的元素均为零, 则关于该图拓扑序列的结论是 ()。

- A 存在, 且唯一 B 存在, 且不唯一 C 存在, 可能不唯一 D 无法确定是否存在

(14) 下列关于图的叙述中, 正确的是 ()。

- 1) 回路是简单路径
 - 2) 存储稀疏图, 用邻接矩阵比邻接表更省空间
 - 3) 若有向图中存在拓扑序列, 则该图不存在回路
- A 仅 2) B 仅 1)、2) C 仅 3) D 仅 1)、3)

(15) 对于有 n 个顶点、 e 条边且使用邻接表存储的有向图进行广度优先遍历, 其算法的时间复杂度是 ()。

- A $O(n)$ B $O(e)$ C $O(n+e)$ D $O(n \times e)$

(16) 下列关于最小生成树的说法中, 正确的是 ()。

- 1) 最小生成树的代价唯一
- 2) 权值最小的边一定会出现在所有的最小生成树中
- 3) 用普里姆 (Prim) 算法从不同顶点开始得到的最小生成树一定相同

4) 使用普里姆和克鲁斯卡尔 (Kruskal) 算法得到的最小生成树总不相同
A 仅 1) B 仅 2) C 仅 1)、3) D 仅 2)、4)

(17) 用哈希 (散列) 方法处理冲突 (碰撞) 时可能出现堆积 (聚集) 现象, 下列选项中, 会受堆积现象直接影响的是 ()。

A 存储效率 B 散列函数 C 装填因子 D 平均查找长度

(18) 对一个长度为 50 的有序表进行折半查找, 最多比较 () 次就能查找出结果。

A 6 B 7 C 8 D 9

(19) 在一棵 m 阶的 B-树种, 若在某结点插入一个新关键字而引起该结点的分裂, 则此结点中原有的关键字的个数是 ()

A m B $m+1$ C $m-1$ D $m/2$

(20) 设二叉排序树关键字由 1-1000 的整数构成, 现要查找关键字为 363 的结点, 下述关键字序列中, 不可能是在二叉排序树上查找的序列是 ()。

A 2, 252, 401, 398, 330, 344, 397, 363
B 924, 220, 911, 244, 898, 258, 362, 263
C 925, 202, 911, 240, 912, 245, 363
D 2, 399, 387, 219, 266, 382, 381, 278, 363

(21) 衡量查找算法性能好坏的主要标准是 ()。

A 参加比较的关键字值的多少
B 被查找的关键字值在关键字序列中的位置
C 关键字值序列中是否存在被查找关键字值
D 关键字值的平均比较次数的多少

(22) 下列排序算法中, () 算法可能出现下面的情况: 在最后一趟开始前, 所有元素都不在最终的位置上。

A 堆排序 B 冒泡排序 C 快速排序 D 插入排序

(23) 基于比较的排序算法时间复杂度最好的是 $O()$ 。

A $\log_2 n$ B n C $n \log_2 n$ D n^2

三、填空题 (本大题共 14 空, 每空 2 分, 本大题共 28 分)

(1) 已知一棵完全二叉树的第 6 层 (设根为第一层) 有 8 个叶子结点, 则该完全二叉树的结点个数最多是_____

(2) 以下程序的功能是实现带附加头结点的单链表数据结点逆序连接，请填空完善之。

```
void reverse(pointer h)
/* h 为附加头结点指针 */
{ pointer p,q;
  p=h->next; h->next=NULL;
  while(_____)
  { q=p; p=p->next; q->next=h->next; h->next=_____; }
}
```

(3) _____是限定仅在表尾进行插入或删除操作的线性表。

(4) 设有一个空栈，栈顶指针为 1000H(十六进制)，现有输入序列为 1, 2, 3, 4, 5。经过 PUSH,PUSH,POP,PUSH,POP,PUSH,PUSH 之后，输出序列是____，而栈顶指针值是____H。设栈为顺序栈，每个元素占 4 个字节。

(5) 已知广义表：A=(a,b), B=(A,A), C=(a,(b,A),B)，求下列运算的结果：tail(head(tail(C)))=(_____)

(6) 已知一棵度为 3 的树有 2 个度为 1 的结点，3 个度为 2 的结点，4 个度为 3 的结点，则该树有_____个叶子结点。

(7) 森林的两类遍历方法为先序遍历与_____；先根遍历树可以借用二叉树的_____遍历算法实现；后根遍历树可以借用二叉树的_____遍历算法实现。

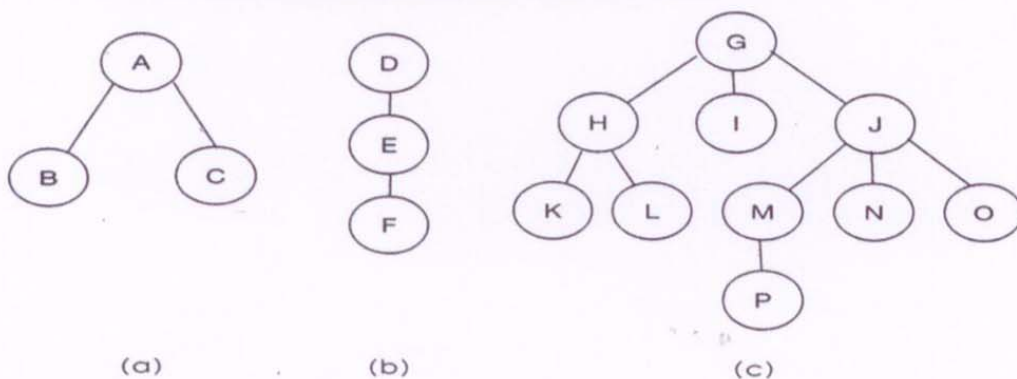
(8) 实现图中广度优先搜索算法时，使用的数据结构是_____

(9) 求解最短路径的 Floyd 算法的时间复杂度为_____

(10) 具有 6 个顶点的无向图，当最少有_____条边时能确保是一个连通图。

四、图示结构题（本大题共 5 小题，每小题 4 分，本大题共 20 分）

(1) 将下图所示的三棵树组成的森林转换为二叉树。

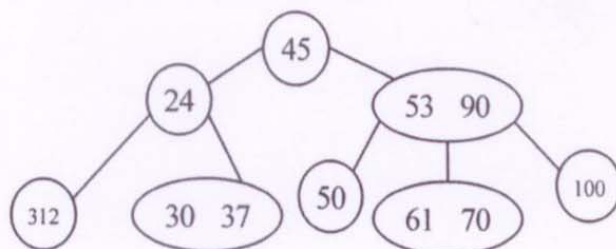


(2) 给定集合{15, 3, 14, 2, 6, 9, 16, 17}。

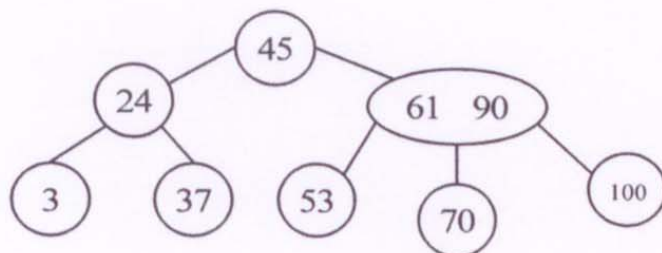
- 1) 用矩形表示外部结点，用圆圈表示内部结点，构造相应的哈夫曼树；
- 2) 计算带权路径长度；
- 3) 写出哈夫曼编码。

(3) 已知 2 棵 2-3 B-树如下 (省略外结点)：

- 1) 对树 (a), 请分别画出先后插入 26, 85 两个新结点后的树形；
- 2) 对树 (b), 请分别画出先后删除 53, 37 两个结点后的树形。

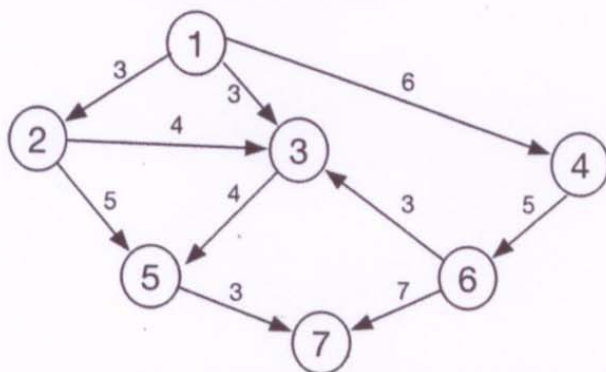


(a)



(b)

(4) 已知一有向图如下图所示



- 1) 写出该图的邻接矩阵表示并据此给出从顶点 1 出发的深度优先遍历序列；
- 2) 求该有向图的强连通分量数目；
- 3) 给出该图的两个拓扑序列。

(5) 设散列函数 $H(K) = 3K \bmod 11$, 散列地址空间为 0-10, 对关键字序列 (32,

13, 49, 24, 38, 21, 4, 12) 按照下述两种解决冲突的方法构造散列表：

1) 线性探测再散列；

2) 链地址法；

3) 并分别求出等概率下查找成功时和查找失败时的平均查找长度 ASL_{succ} 和 ASL_{unsucc} 。

五、阅读以下函数、指出算法的功能（本大题共 5 小题，每小题 4 分，本大题共 20 分）

(1) Status A1(LinkList &L,int n) // L 为单链线性表

```
{
    int j;
    LinkList p,q,s;
    if(n<=0)
        return ERROR;
    InitList(L);
    printf("请输入%d 个元素:\n",n);
    s=(LinkList)malloc(sizeof(LNode));
    scanf("%d",&s->data);
    s->next=NULL;
    L->next=s;
    for(j=1;j<n;j++)
    {
        s=(LinkList)malloc(sizeof(LNode));
        scanf("%d",&s->data);
        q=L;
        p=L->next;
        while(p->data<s->data)
        {
            q=p;
            p=p->next;
        }
        s->next=q->next;
        q->next=s;
    }
    return OK;
}
```

(2) Status A2(LinkQueue &Q,QElemType &e) //Q 为链队列


```

{
    QueuePtr p;
    if(Q.front==Q.rear)
        return ERROR;
    p=Q.front->next;
    e=p->data;
    Q.front->next=p->next;
    if(Q.rear==p)
        Q.rear=Q.front;
    free(p);
    return OK;
}

```

```

(3)
#define MAXSIZE 20
typedef int KeyType;
typedef int InfoType;
typedef SqList B;
struct RedType
{
    KeyType key;
    InfoType otherinfo;
};

struct SqList
{
    RedType r[MAXSIZE+1];
    int length;
};

void F(B &H,int s,int m)
{
    RedType rc;
    int j;
    rc=H.r[s];
    for(j=2*s;j<=m;j*=2)
    {
        if(j<m&&LT(H.r[j].key,H.r[j+1].key))

```



```

        ++j;
        if(!LT(rc.key,H.r[j].key))
            break;
        H.r[s]=H.r[j];
        s=j;
    }
    H.r[s]=rc;
}
void A3(B &H)
{
    RedType t;
    int i;
    for(i=H.length/2;i>0;--i)
        F(H,i,H.length);
    for(i=H.length;i>1;--i)
    {
        t=H.r[1];
        H.r[1]=H.r[i];
        H.r[i]=t;
        F(H,1,i-1);
    }
}

```

(4) Status A4(BiThrTree T,Status(*Visit)(TElemType e)) //T 为二叉树的二叉线索存储结构

```

{
    BiThrTree p;
    p=T->lchild;
    while(p!=T)
    {
        while(p->LTag==Link)
            p=p->lchild;
        if(!Visit(p->data))
            return ERROR;
        while(p->RTag==Thread&& p->rchild!=T)
        {
            p=p->rchild;
            Visit(p->data);
        }
    }
}

```

```

    }
    p=p->rchild;
}
return OK;
}

```

(5) void A5(BiTree &p) //BiTree 为二叉树的二叉链表存储表示

```

{
    BiTree q,s;
    if(!p->rchild)
    {
        q=p;
        p=p->lchild;
        free(q);
    }
    else if(!p->lchild)
    {
        q=p;
        p=p->rchild;
        free(q);
    }
    else
    {
        q=p;
        s=p->lchild;
        while(s->rchild)
        {
            q=s;
            s=s->rchild;
        }
        p->data=s->data;
        if(q!=p)
            q->rchild=s->lchild;
        else
            q->lchild=s->lchild;
        free(s);
    }
}
}

```

六、算法设计题（本大题共 3 小题， 每小题 10 分， 本大题共 30 分）

（1）编写算法求有向图 G 中距离顶点 v_0 简单路径长度为 len 的所有顶点。

（2）二叉树的带权结点路径长度（WPL）是二叉树中所有叶子结点的带权路径长度之和。给定一棵二叉树 T ，采用二叉链表存储，结点结构为

left	weight	right
------	--------	-------

其中叶子结点的 `weight` 域保存该结点的非负权值。设 `root` 为指向 T 的根结点的指针，请设计求 T 的 WPL 的算法，要求：

- 1) 给出算法的基本设计思想；
- 2) 使用 C 或 C++ 语言，给出二叉树结点的数据类型定义；
- 3) 根据设计思想，采用 C 或 C++ 语言描述算法，关键之处给出注释。

（3）设有一个数组中存放了一个无序的关键序列 K_1, K_2, \dots, K_n 。现要求将 K_n 放在元素排序后的正确位置上，试编写实现该功能的算法，要求比较关键字的次数不超过 n 。