

华中师范大学 2012-2013 学年第 1 学期  
期末考试试卷 (A 卷)

课程名称 数据结构 课程编号 84810007 任课教师 魏开平、王敬华、沈显君

题型	填空题	单选题	计算题	编程题		总分
分值	30	10	30	20		100
得分						

得分	评阅人

一、填空 (每个空格 2 分, 共 30 分)

1. 数据结构被形式地定义为  $(D, R)$ , 其中  $D$  是数据元素的有限集合,  $R$  是  $D$  上的有限集合。
2. 栈是一种特殊的线性表, 允许插入和删除运算的一端称为\_\_\_\_\_。
3. \_\_\_\_\_是被限定为只能在表的一端进行插入运算, 在表的另一端进行删除运算的线性表。
4. 计算广义表:  $\text{GetHead}(\text{GetTail}(\text{GetHead}(((a,b),(c,d)))) = \underline{\hspace{2cm}}$ 。
5. 若对其进行快速排序, 在最坏的情况下所需要的时间是\_\_\_\_\_。
6. 三元素组表中的每个结点对应于稀疏矩阵的一个非零元素, 它包含有三个数据项, 分别表示该元素的\_\_\_\_\_。
7. 一棵具有 257 个结点的完全二叉树, 它的深度为\_\_\_\_\_。
8. 用 5 个权值  $\{3, 2, 4, 5, 1\}$  构造的哈夫曼(Huffman)树的带权路径长度是\_\_\_\_\_。
9. 设一棵完全二叉树有 700 个结点, 则共有\_\_\_\_\_个叶子结点。
10. 若要求一个稠密图  $G$  的最小生成树, 最好用\_\_\_\_\_算法来求解。
11. 设有一稀疏图  $G$ , 则  $G$  采用\_\_\_\_\_存储较省空间。
12. 在各种查找方法中, 平均查找长度与结点个数  $n$  无关的查找方法是\_\_\_\_\_查找。
13. 散列法存储的基本思想是由\_\_\_\_\_决定数据的存储地址。
14. 在堆排序和快速排序中, 若初始记录基本无序, 则最好选用\_\_\_\_\_排序。
15. 对于  $n$  个记录的集合进行归并排序, 所需要的附加空间是\_\_\_\_\_。



得分	评阅人

## 二、单选题。(每小题1分,共10分)

- 算法分析的目的是\_\_\_\_。  
 A) 找出数据结构的合理性  
 B) 研究算法中的输入和输出的关系  
 C) 分析算法的效率以求改进  
 D) 分析算法的易懂性和文档性
- 栈中元素的进出原则是\_\_\_\_。  
 A) 先进先出  
 B) 后进先出  
 C) 栈空则进  
 D) 栈满则出
- 数组  $Q[n]$  用来表示一个循环队列,  $f$  为当前队列头元素的前一位置,  $r$  为队尾元素的位置, 假定队列中元素的个数小于  $n$ , 计算队列中元素多少的公式为\_\_\_\_。  
 A)  $r-f$   
 B)  $(n+r-f)\%n$   
 C)  $n+r-f$   
 D)  $(n+f-r)\%n$
- 设有两个串  $p$  和  $q$ , 求  $q$  在  $p$  中首次出现的位置的运算称作\_\_\_\_。  
 A) 模式匹配  
 B) 连接  
 C) 求子串  
 D) 求串长
- 串是一种特殊的线性表, 其特殊性体现在\_\_\_\_。  
 A) 可以顺序存储  
 B) 数据元素可以是多个字符  
 C) 可以链式存储  
 D) 数据元素是一个字符
- 图的深度优先遍历类似于二叉树的\_\_\_\_。  
 A) 先序遍历  
 B) 中序遍历  
 C) 后序遍历  
 D) 层次遍历
- 有 8 个结点的无向连通图最少有\_\_\_\_条边。  
 A) 5  
 B) 6  
 C) 7  
 D) 8
- 对 22 个记录的有序表作折半查找, 当查找失败时, 至少需要比较\_\_\_\_次关键字。  
 A) 3  
 B) 4  
 C) 5  
 D) 6
- 链表适用于\_\_\_\_查找。  
 A) 顺序  
 B) 二分法  
 C) 顺序, 也能二分法  
 D) 随机
- 堆的形状是一棵\_\_\_\_。  
 A) 二叉排序树  
 B) 满二叉树  
 C) 完全二叉树  
 D) 二叉树

得分	评阅人

### 三. 计算题。(每小题8分, 共40分)

26. 对于给定的一组权  $w=\{1,4,9,16,25,36,49,64,81\}$ , 请构造一棵哈夫曼树, 并求它的带权路径长度。

27. 已知一组数据的输入顺序为: 53,78,65,17,87,09,81,15, 试为该序列建立二叉查找树, 写出该二叉查找树, 并求出该树的平均查找长度。

28. 设哈希(Hash)表的地址范围为  $0 \sim 17$ , 哈希函数为:

$$H(K)=K \text{ MOD } 16.$$

K 为关键字, 用线性探测法再散列法处理冲突, 输入关键字序列: (10, 24, 32, 17, 31, 30, 46, 47, 40, 63, 49), 试回答下列问题:

- (1) 画出哈希表的示意图;
- (2) 若查找关键字 63, 需要依次与哪些关键字进行比较?
- (3) 若查找关键字 60, 需要依次与哪些关键字比较?
- (4) 假定每个关键字的查找概率相等, 求查找成功时的平均查找长度。



得分	评阅人

### 三. 计算题。(每小题8分, 共40分)

26. 对于给定的一组权  $w=\{1,4,9,16,25,36,49,64,81\}$ , 请构造一棵哈夫曼树, 并求它的带权路径长度。

27. 已知一组数据的输入顺序为: 53,78,65,17,87,09,81,15, 试为该序列建立二叉查找树, 写出该二叉查找树, 并求出该树的平均查找长度。

28. 设哈希(Hash)表的地址范围为  $0 \sim 17$ , 哈希函数为:

$$H(K)=K \text{ MOD } 16.$$

K 为关键字, 用线性探测法再散列法处理冲突, 输入关键字序列: (10, 24, 32, 17, 31, 30, 46, 47, 40, 63, 49), 试回答下列问题:

- (1) 画出哈希表的示意图;
- (2) 若查找关键字 63, 需要依次与哪些关键字进行比较?
- (3) 若查找关键字 60, 需要依次与哪些关键字比较?
- (4) 假定每个关键字的查找概率相等, 求查找成功时的平均查找长度。

得分	评阅人

四. 编程题。(每小题 10 份, 共 20 分)

31. 设  $h$  为带表头结点的循环链表的头指针, 请编写一个删除表中数据域值为  $x$  的所有结点的算法。

注: 链表结点类型如下

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

```
};
```

算法函数说明: `void del_x( Node *h, int x );`



32. 编写递归算法, 求在二叉树中位于先序序列中第  $k$  个位置的结点的值。

注: 二叉树结点类型如下

```
struct Bitree {  
    int data;  
    struct Bitree *lchild, *rchild;
```

```
};
```

算法函数说明: `int GetPreSeq( Bitree *T );`

### 一、填空题

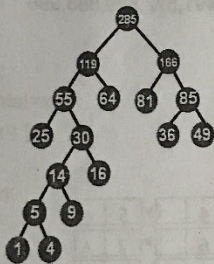
1. 元素的序偶
2. 栈顶
3. 队列
4. (c,d)
5.  $O(n^2)$
6. 行、列、元素值
7. 9
8.  $(1+2)*3+(3+4+5)*2=33$
9. 350
10. Prim
11. 邻接表
12. HASH
13. 关键字
14. 快速
15.  $O(n)$

## 二、单选题

1. C      2. B      3. D      4. A      5. D  
6. A      7. B      8. B      9. A      10. C

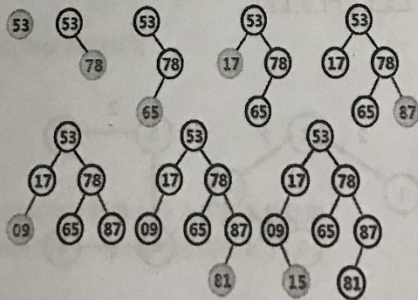
### 三、计算题

1. 解



$$wpl = (1+4)*6 + 9*5 + 16*4 + 25*3 + 64*2 + (36+49)*3 + 81*2 = 759$$

2. 解



$$ASL = (1 + 2 \cdot 2 + 3 \cdot 3 + 4 \cdot 2) / 8 = 17/8$$

- ### 3. 解



(1) 哈希表

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
32	17	63	49					24	40	10				30	31	46	47

(2) 查找 63, 首先要与  $H(63)=63\%16=15$  号单元内容比较, 即 63 vs 31, no; 然后顺移,

与 46, 47, 32, 17, 63 相比, 一共比较了 6 次!

(3) 查找 60, 首先要与  $H(60)=60\%16=12$  号单元内容比较, 但因为 12 号单元为空 (应当有空标记), 所以应当只比较这一次即可。

(4) 对于黑色数据元素, 各比较 1 次; 共 6 次; 对红色元素则各不相同, 要统计移位的位数。"63"需要 6 次, "49"需要 3 次, "40"需要 2 次, "46"需要 3 次, "47"需要 3 次, 所以  $ASL=1/11(6+2+3\times3)=17/11\approx1.55$

4. 解

初始: 491, 77, 572, 16, 996, 101, 863, 258, 689, 325

第 1 趟: [325, 77, 258, 16, 101] 491 [863, 996, 689, 572]

第 2 趟: [101, 77, 258, 16] 325, 491 [863, 996, 689, 572]

第 3 趟: [16, 77] 101 [258] 325, 491 [863, 996, 689, 572]

第 4 趟: 16 [77] 101 [258] 325, 491 [863, 996, 689, 572]

第 5 趟: 16, 77, 101 [258] 325, 491 [863, 996, 689, 572]

第 6 趟: 16, 77, 101, 258, 325, 491 [863, 996, 689, 572]

第 7 趟: 16, 77, 101, 258, 325, 491 [572, 689] 863 [996]

第 8 趟: 16, 77, 101, 258, 325, 491, 572 [689] 863 [996]

第 9 趟: 16, 77, 101, 258, 325, 491, 572, 689, 863 [996]

第 10 趟: 16, 77, 101, 258, 325, 491, 572, 689, 863, 996

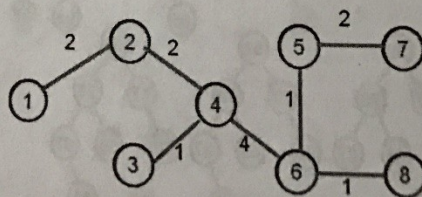
5. 解:

(1) 邻接表:

1	→	1	→	2	∧				
2	→	0	→	2	∧				
3	→	0	→	3	∧				
4	→	1	→	2	→	4	→	5	∧
5	→	3	→	5	→	6	→	7	∧
6	→	3	→	4	→	6	→	7	∧
7	→	4	→	5	→	7	∧		
8	→	4	→	5	→	6	∧		

(2) 1, 2, 4, 3, 5, 6, 7, 8

(3)



四、编程题



1. 解

```
void del_x( Node *h, int x )
{
    Node *p,*q;
    q=h; p=h->link;
    while(p!=h) {
        if (p->data==x) {
            q->next=p->next;
            free(p);
            p=q->next;
        }
        else { q=p; p=p->next; }
    }
}

void LkListDelete( LinkList &La, int x )
{
    if (!La) Error("UnderFlow");
    if ( La->data==x ) {
        p=La;
        La=La->next;
        free(p);
    }
    else {
        q=La; p=La->next;
        while ( p && p->data!=x )
            if ( p->data!=x ) {
                q=p;
                p=p->next;
            }
        if ( p ) {
            q->next=p->next;
            free(p);
        }
        else Print("Not Found");
    }
}
```

2. 解

```
int c=0, k; //这里把 k 和计数器 c 作为全局变量处理
int GetPreSeq(Bitree *T) //求先序序列第 k 个结点
{
    if (T) {
        c++; //每访问一个子树的根都会使前序序号计数器加 1
```

}