

2018-2019 学年第一学期《数据结构》期末考试试卷(B 卷)

参考答案及评分标准

一、单项选择题(本大题共 10 小题, 每小题 1 分, 共 10 分)

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

二、简答题(本大题共 2 小题, 每小题 5 分, 共 10 分)

1. 简述顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好？

答：①顺序存储时，相邻数据元素的存放地址也相邻（逻辑与物理统一）；要求内存中可用存储单元的地址必须是连续的。 1 分

优点：元素可随机存取，存储单元利用率高。 0.5 分

缺点：插入或删除元素时不方便，需移动大量元素。 0.5 分

②链式存储时，相邻数据元素可随意存放，但所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针。 1 分

优点：插入或删除元素时很方便，使用灵活。 0.5 分

缺点：元素不可随机存取，存储单元利用率低。 0.5 分

顺序表适宜于做查找这样的静态操作；链表宜于做插入、删除这样的动态操作。若线性表的长度变化不大，且其主要操作是查找，则采用顺序表；若线性表的长度变化较大，且其主要操作是插入、删除操作，则采用链表。 1 分

2. 对于 m 阶的 B-树，请简述其特性。对于高度为 3 的 3 阶 B-树最少的关键字总数是多少？

答：一棵 m 阶 B-树，或为空树，或为满足下列特性的树：

(1)树中每个结点至多有 m 棵子树； 0.5 分

(2)若根结点不是叶子结点，则至少有两棵子树； 0.5 分

(3)除根之外的所有非终端结点至少有 $\lceil m/2 \rceil$ 棵子树； 1 分

(4)所有非终端结点包含下列信息数据： $(n, A_0, K_1, A_1, K_2, A_2, \dots, K_n, A_n)$ ，其中 $K_i (i=1, \dots, n)$ 为关键字，且 $K_i < K_{i+1} (i=1, \dots, n-1)$ ； $A_i (i=0, \dots, n)$ 为指向子树根结点的指针，且 A_{i-1} 所指子树中所有结点的关键字均小于 $K_i (i=1, \dots, n)$ ， A_n 所指子树中所有结点的关键字均大于 K_n 。 $n(\lceil m/2 \rceil - 1) = < n <= m - 1$ 为关键字的总数； 1 分

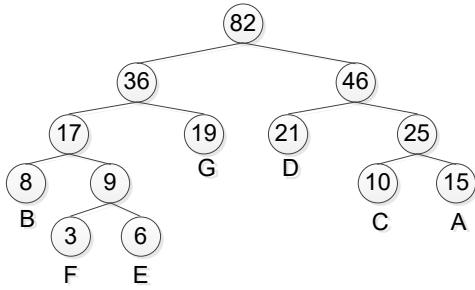
(5)所有叶子结点都出现在同一层，并且不带信息。 1 分

对于高度为 3 的 3 阶 B-树最少的关键字总数是 7 个。 1 分

三、综合应用题(本大题共 6 小题, 每小题 10 分, 共 60 分)

1. 给定一组数列 (15, 8, 10, 21, 6, 3, 19) 分别代表字符 A、B、C、D、E、F、G 出现的频度：

(1) 构造哈夫曼树，计算其 WPL 值；(6 分)

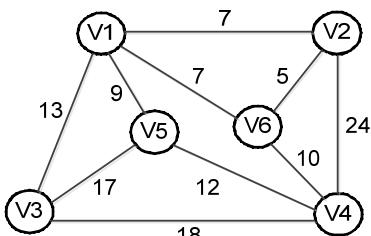


(注: 此答案不唯一) 4 分

$$WPL = 19*2 + 21*2 + 8*3 + 10*3 + 15*3 + 3*4 + 6*4 = 215 \text{ 2 分}$$

(2) 给出各字符的哈夫曼编码。(4分)

A: 111 B: 000 C: 110 D: 10 E: 0011 F: 0010 G: 01 (注: 此答案依赖(1)所构造哈夫曼树)----4分



2. 已知一个无向图 G:

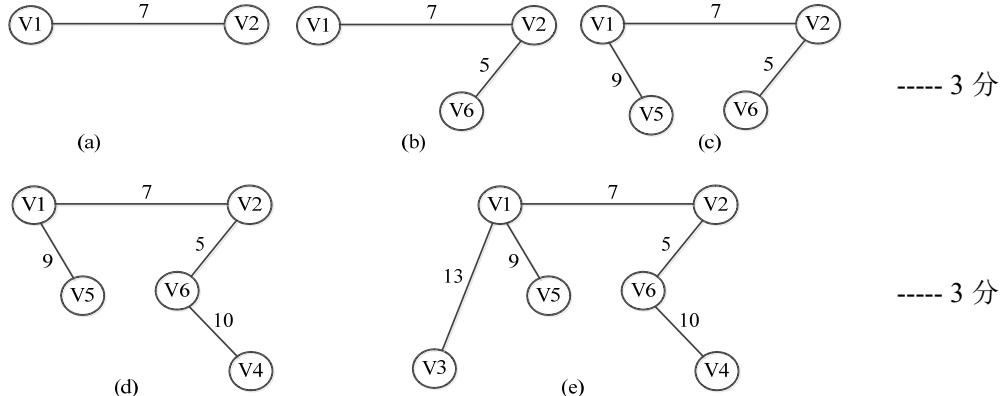
(1) 给出以顶点 V1 为起始点的深度优先搜索序列和广度优先搜索序列; (4分)

(注: 此答案不唯一!)

以顶点 V1 为起始点的深度优先搜索序列: V1, V2, V4, V3, V5, V6 -- 2 分

以顶点 V1 为起始点的广度优先搜索序列: V1, V2, V3, V5, V6, V4 -- 2 分

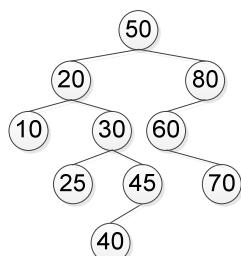
(2) 给出用 Prim 算法构造最小生成树的过程 (以顶点 V1 为起始点)。 (6分)



(注: 此答案不唯一!)

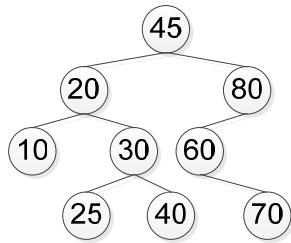
3. 若关键字输入顺序为 (50, 20, 30, 25, 80, 10, 45, 60, 70, 40)。

(1) 构造二叉排序树; (4分)



..... 4 分

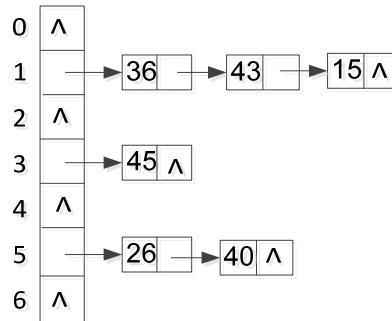
(2) 给出删除关键字 50 后的二叉排序树。 (6 分)



(注: 此答案不唯一!).....6 分

4. 已知一组关键字为 (26, 36, 40, 43, 45, 15), 用链地址法解决冲突, 哈希函数的形式为 $H(K) = K \text{ MOD } 7$ 。

(1) 构造哈希表; (6 分)



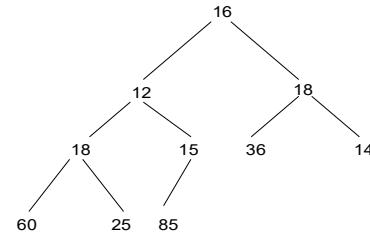
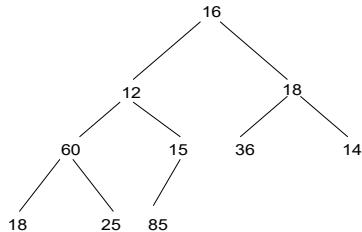
-----6 分

(2) 计算出等概率情况下查找成功的平均查找长度 ASL_{succ} 。 (4 分)

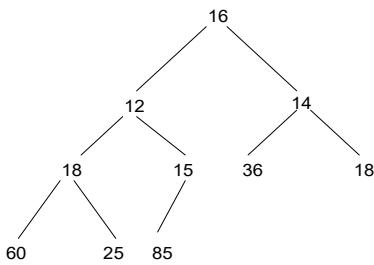
$$ASL_{\text{succ}} = (3*1 + 2*2 + 1*3)/6 = 10/6 \quad \text{-----4 分}$$

5. 已知一组待排记录的关键字序列为(16, 12, 18, 60, 15, 36, 14, 18, 25, 85), 用堆排序方法建小根堆, 请解答以下 2 个问题:

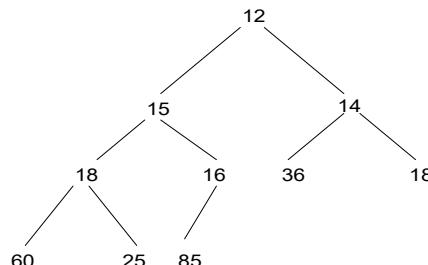
(1) 画出建初始堆的过程; (8 分)



(1)-----2 分



(2)-----2 分



(3)-----2 分

(2) 给出建堆后的关键字序列。 (2 分)

12, 15, 14, 18, 16, 36, 18, 60, 25, 85

6. 对下面关键字序列，写出采用希尔排序算法进行排序的每一趟的结果。其中增量序列的取值为{ 5, 3, 1 }，其关键字序列为 (125, 11, 22, 34, 15, 44, 76, 66, 100, 8, 14, 20, 2, 5, 1) (10 分)。

第一趟: 14 11 2 5 1 44 20 22 34 8 125 76 66 100 15 ---- (4 分)
 第二趟: 5 1 2 8 11 15 14 22 34 20 100 44 66 125 76---- (4 分)
 第三趟: 1 2 5 8 11 14 15 20 22 34 44 66 76 100 125---- (2 分)

四、算法设计题(本大题共 2 小题，每小题 10 分，共 20 分)

1. 已知带头结点的线性表采用链式结构存储，请写出适当的其结点定义并请设计算法，将带头结点的单链表中所有元素结点逆转倒排列。(注：要有适当的注释！)

```
Typedef struct node {
    ElemType data; //数据域----- 0.5 分
    struct node *next; //指针域----- 0.5 分
}LinkNode, *LinkList; ----- 1 分
LinkList reverse(LinkList head) {
    LinkList p, q; ----- 0.5 分
    p=head->next; ----- 0.5 分
    head->next=NULL; ----- 1 分
    while(p!=NULL) { //头插法实现单链表元素结点逆转----- 1 分
        q=p;
        p=p->next; ----- 1 分
        q->next=head->next; ----- 1 分
        head->next=q; ----- 1 分
    }
    return head; //返回成功逆转后单链表的头指针 ----- 1 分
}
```

2. 已知二叉树采用二叉链表存储，其结点结构定义如下：

```
Typedef struct TreeNode{
    ElmType data;
    struct TreeNode *lchild, *rchild;
}*BiTree;
请编写递归函数 int SumNodes(BiTree T)，返回二叉树 T 的结点总数。
int SumNodes(BiTree T)-----1 分
{
    if(T==NULL) -----1 分
    return 0; -----1 分
    return SumNodes((T->lchild))+SumNodes(T->rchild)+1; -----5 分
}
```