2018-2019 学年第一学期《数据结构》期末考试试卷(A 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、D　3、B　4、D　5、B　6、B　7、A　8、C　9、B　10、C

二、简答题(本大题共 2 小题，每小题 5 分，共 10 分)

1. 简述线性表、栈与队的异同点。

答：**相同点：**①都是线性结构，都是逻辑结构的概念，都可以用顺序存储或链式存储来实现；　---1 分

②栈和队列是两种特殊的线性表，即操作受限的线性表，只是对插入、删除操作位置加以限制。　---1 分

不同点：①运算规则不同：线性表为随机存取，而栈是只允许在表的一端进行插入、删除操作，是后进先出表 LIFO；　---1 分

队列是只允许在表的一端进行插入操作而在另一端进行删除操作，是先进先出表 FIFO。　---1 分

②用途不同：栈用于子程调用和保护现场，队列用于多道作业处理、指令寄存及其他运算等。　---1 分

2．简述以下三个概念的区别：首元结点、头结点和头指针。在单链表中设置头结点的作用是什么？

答：首元结点是指链表中存放第一个数据元素 a_1 的结点。　------1 分

头结点是指在链表的首元结点之前附设的一个结点，该节点数据域内只放空表标志和表长等信息。------1 分

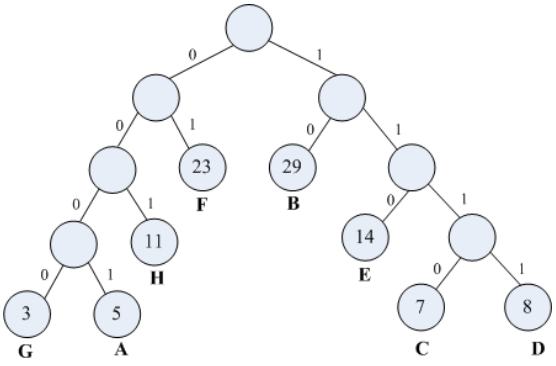
头指针是指指向链表中第一个结点（或为头结点或为首元结点）的指针。　-----1 分

在单链表中设置头结点的作用是为了对链表进行操作时，可以对空表、非空表的情况以及对首元结点进行统一处理。----------2 分

三、综合应用题(本大题共 6 小题，每小题 10 分，共 60 分)

1. 已知某系统在通信联络中只可能出现 8 种字符：A、B、C、D、E、F、G、H，其概率分别为 0.05、0.29、0.07、0.08、0.14、0.23、0.03、0.11，请解答以下 2 个问题：

(1) 假设各字符权重为 w={5，29，7，8，14，23，3，11}，请构造哈夫曼树，并计算其 WPL 值；(6 分)



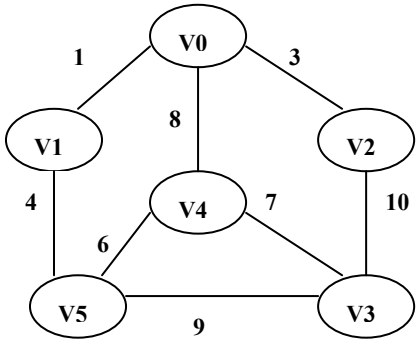
(注：此答案不唯一)----------4 分

WPL=23×2+29×2+11×3+14×3+3×4+5×4+7×4+8×4=271　----------2 分

(2) 在(1)构造的哈夫曼树的基础上，请给出各字符的哈夫曼编码。(4 分)

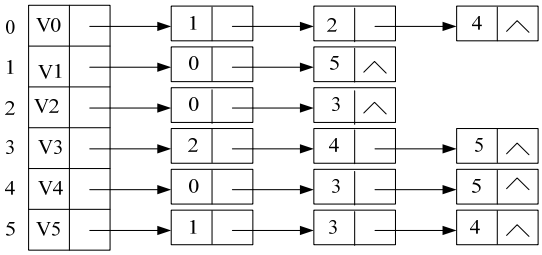
A:0001 B:10 C:1110 D:1111 E:110 F:01 G:0000 H:001 (注：此答案依赖于(1)所构造的哈夫曼树) ----------4 分

2. 对于下图，请解答以下 3 个问题：



(1) 按照链表各结点序号递增原则，画出该图的邻接链表；(4 分)

答：



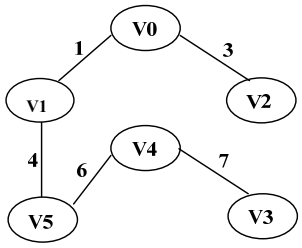
----------4 分

(2) 根据(1)所构造的邻接链表，以 V0 为出发点，给出它的广度优先遍历序列；(2 分)

答：v0 v1 v2 v4 v5 v3 ----------2 分

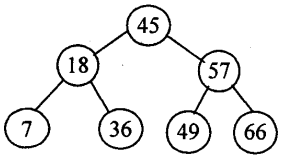
(3) 以 V0 为出发点，利用普里姆算法，画出它的最终最小生成树。(4 分)

答：

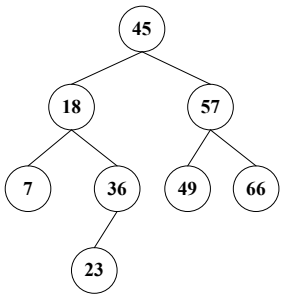


----------4 分

3．已知一棵二叉排序树如图所示，请解答以下 3 个问题：

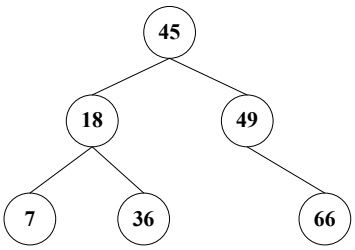


(1) 画出插入元素 23 后此二叉排序树的结构；(2 分)



----------2 分

(2) 画出在原树中删除元素 57 后此二叉排序树的结构；(5 分)



(注：此答案不唯一)　----------5 分

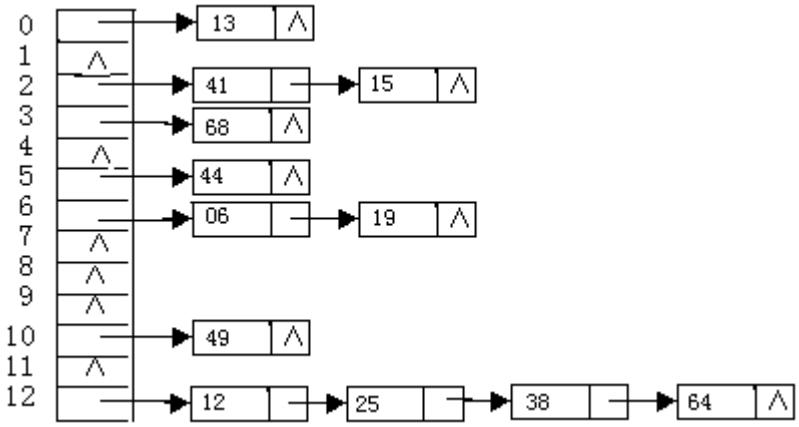
(3) 假设二叉排序树的 RDL 遍历算法定义如下：若二叉排树非空，则依次执行如下操作：①遍历右子树；②访问根节点；③遍历左子树。请给出原树 RDL 遍历的结果序列 。(3 分)

66　57　49　45　36　18　7　　----------3 分

4. 已知一组关键字(13，41，15，44，06，68，12，25，38，64，19，49)，哈希函数为 H（K）=K MOD 13，采用链地址法处理冲突，请解答以下 2 个问题：

(1)画出相应的哈希表；(7 分)

答：13%13 = 0; 41%13 = 2; 15%13 = 2; 44%13 = 5; 6%13 = 6; 68%13 = 3; 12%13 = 12;
25%13 == 12; 38%13 = 12; 64%13 = 12; 19%13 = 6; 49%13 = 10;

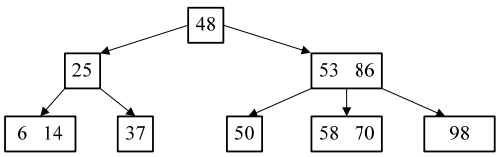


--------7 分

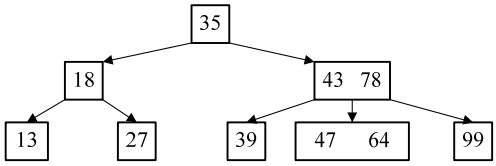
(2)等概率情况下，查找成功的平均查找长度 ASL_{succ}。(3 分)

$$ASL_{succ}=\frac{1}{12}(7*1+3*2+1*3+1*4)=\frac{20}{12}$$　　----------3 分

5. 已知如下图 2 棵 B-树，请解答以下 2 个问题：

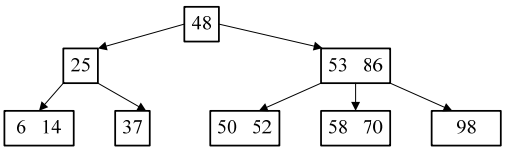


(a)3 阶 B-树

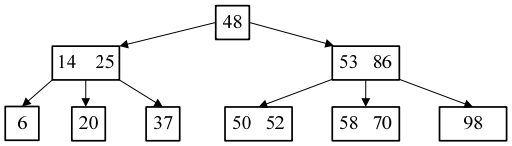


(b)4-阶 B-树

(1) 请依次画出插入 52 和 20 后的 3 阶 B-树。(5 分)

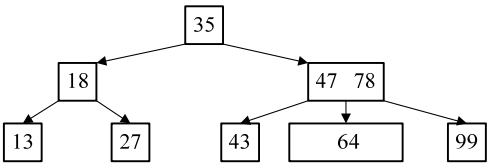


插入 52 后的 3 阶 B-树 ---2 分

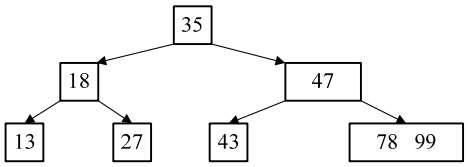


插入 20 后的 3 阶 B-树 ---3 分

(2) 请依次画出删除 39 和 64 后的 4 阶 B-树。(5 分)



删除 39 后的 4 阶 B-树 ---2 分

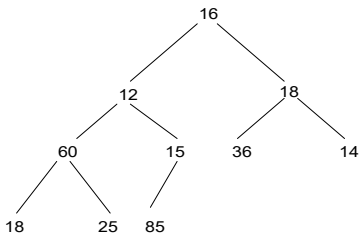


删除 64 后的 4 阶 B-树

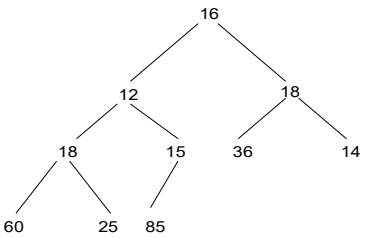
(注：此答案不唯一！)----3 分

6．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：

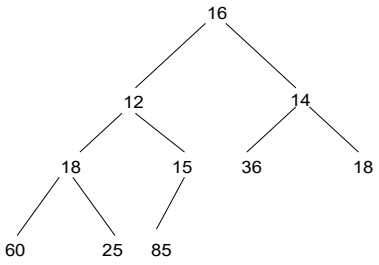
(1) 画出建初始堆的过程； (8 分)



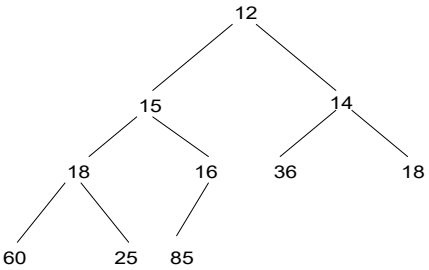
(1)----------2 分



(2)----------2 分



(3)---------2 分



(4)----------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

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，其结点的定义如下：

```
Typedef struct LinearNode{
        ElmType data;
        struct Node    *next;
}*List;
```

请设计算法 int DeleteNodes(List L, ElmType e)，删除结点的值为 e 的所有结点，返回被删除结点的个数。(注：要有适当的注释！)

```
int DeleteNodes(List L, ElmType e)                                    ----0.5 分
{
    int num=0;                                                        ----0.5 分
```

```
    List p, q;                                                          ----0.5 分
    q=L;                                                                ----0.5 分
    p=L->next;                                                          ----0.5 分
    while(p)     /*检查链表中每一个结点*/                                  ----0.5 分
     {
        if(p->data==e)              /*找到了元素值为 e 的结点*/           ----0.5 分
         {
           num++;                     /*被删结点个数加 1*/               ----0.5 分
           q->next=p->next;     /*  删除结点 p */                       ----1 分
           free(p);                  /*  释放被删除元素的空间  */         ----1 分
           p=q;           /*  修改 p 的值，使其指向被删结点的后继结点  */  ----1 分
         }
        else
        {
          q=p;                       /* q 指向 p 所指向的结点  */         ----1 分
          p=p->next;                 /* p 指向其后继结点  */              -----1 分
         }
      }
  return num;                                                           ----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
{                                   ----------1 分
    if(T==NULL) ----------1 分
        return 0; ----------1 分
    return SumNodes((T->lchild))+SumNodes(T->rchild)+1; ----------5 分
}                                       ----------1 分
```