

2019-2020 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、A 2、C 3、B 4、B 5、D 6、C 7、A 8、A 9、C 10、B

二、简答题(本大题共 2 小题, 每小题 5 分, 共 10 分)

1. 请简述算法的定义及其特性。

答：算法是对特定问题求解步骤的一种描述，是指令的有限序列。-----1 分

算法的五大特性如下：

(1) 有穷性：一个算法必须总是在执行有穷步之后结束，且每一步都在有穷时间内完成。
-----1 分

(2) 确定性：算法中的每一个步骤必须有确定含义，无二义性，对于相同的输入只能得到相同的输出。
-----1 分

(3) 可行性：原则上能精确进行，操作可通过已实现的基本运算执行有限次而完成。
-----1 分

(4) 输入：有多个或 0 个输入。
-----0.5 分

(5) 输出：至少有一个或多个输出。
-----0.5 分

2. 对于 m 阶的 B-树，请简述其特性。

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；-----1 分

(2) 若根结点不是叶子结点，则至少有两棵子树；---1 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

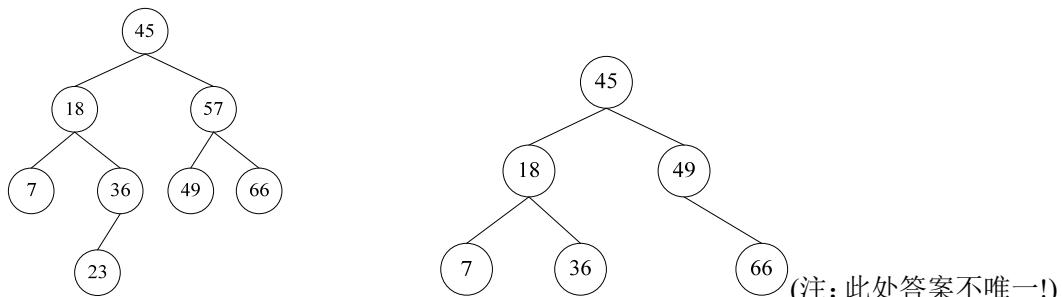
(4) 所有非终端结点包含下列信息数据： $(n, A_0, K_1, A_1, K_2, A_2, \dots, K_n, A_n)$ ，其中 $K_i (i=1, \dots, n)$ 为关键字，且 $K_i < K_{i+1} (i=1, \dots, n-1)$ ； $A_i (i=0, \dots, n)$ 为指向子树根结点的指针，且 A_{i-1} 所指子树中所有结点的关键字均小于 $K_i (i=1, \dots, n)$ ， A_n 所指子树中所有结点的关键字均大于 K_n ， $n (\lceil m/2 \rceil - 1 \leq n \leq m - 1)$ 为关键字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

(5) 所有叶子结点都出现在同一层，且不带信息。
-----1 分

三、应用题(本大题共 6 小题, 每小题 10 分, 共 60 分)

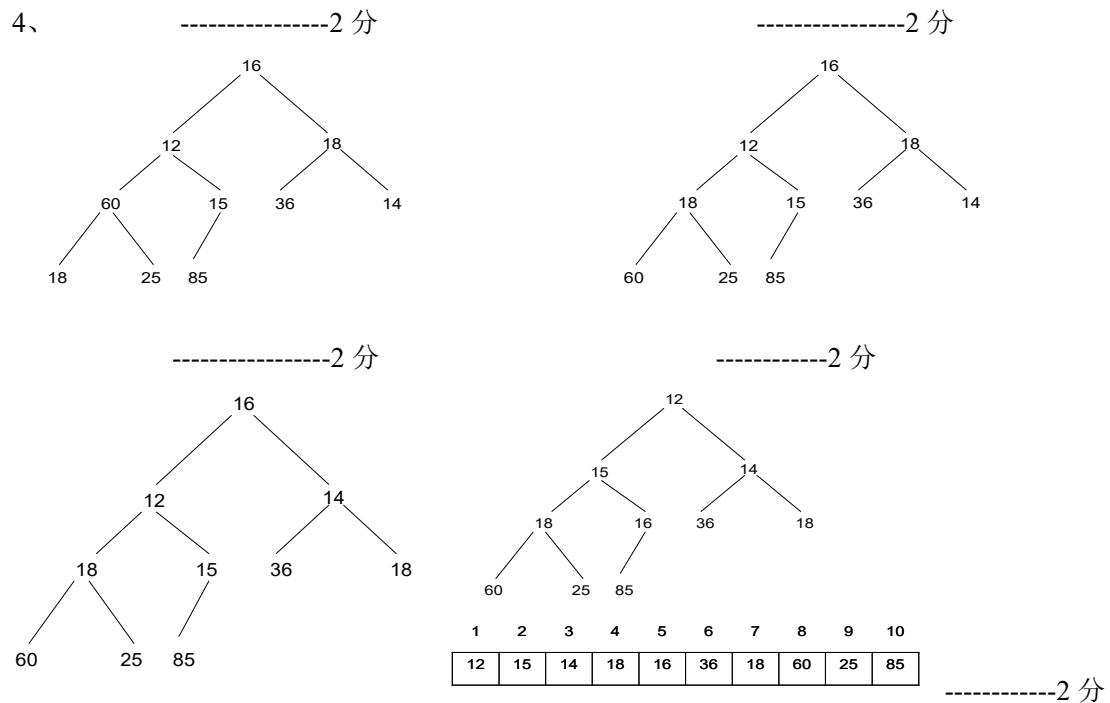
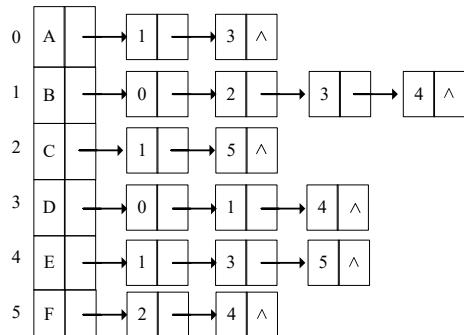
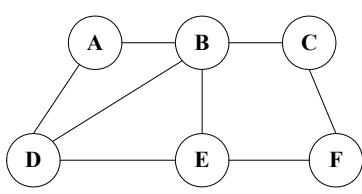
1、遍历结果序列：G C F A B D-----10 分

2、(1) 插入元素 23 后二叉排序树结构----4 分 (2) 原树删除 57 元素后二叉排序树结构----6 分

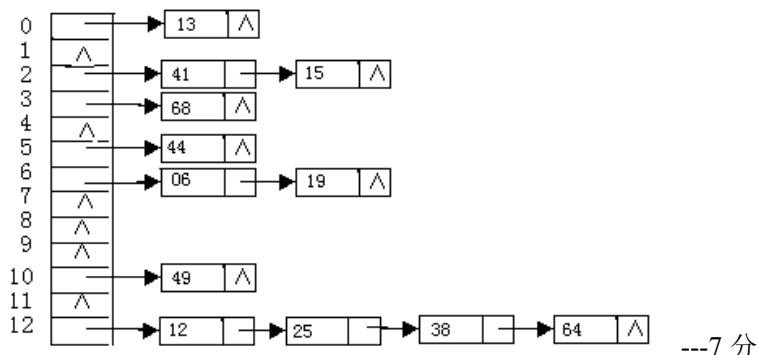


3、(1) 对应的图 G -----5 分

(2) G 的邻接表存储结构-----5 分



5、(1) $13\%13 = 0$; $41\%13 = 2$; $15\%13 = 2$; $44\%13 = 5$; $6\%13 = 6$; $68\%13 = 3$; $12\%13 = 12$;
 $25\%13 = 12$; $38\%13 = 12$; $64\%13 = 12$; $19\%13 = 6$; $49\%13 = 10$;

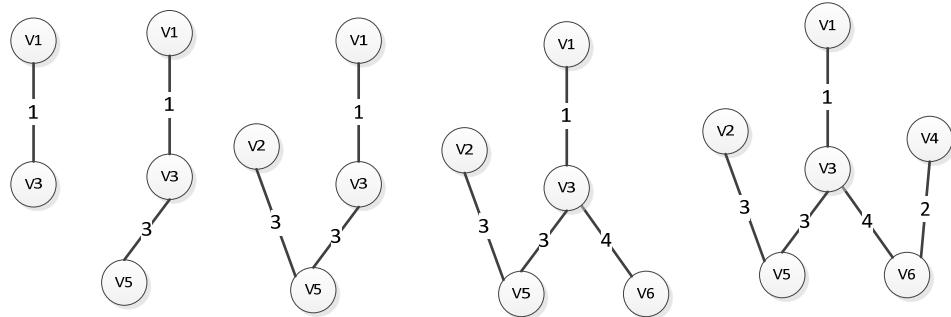


$$(2) \quad ASL_{succ} = \frac{1}{12}(7*1 + 3*2 + 1*3 + 1*4) = \frac{20}{12} \quad \text{-----3分}$$

6、(1) 图的广度优先搜索序列为: V1 V2 V3 V4 V5 V6 -----2分

深度优先搜索序列为: V1 V2 V3 V4 V6 V5 -----2分(注: 此处答案不唯一!)

(2) 根据 prim 算法, 求图 G 从顶点 V1 出发的最小生成树, 要求表示出其每一步生成过程, 用图的方式表示。--6 分



四、算法设计题(本大题共 2 小题, 每小题 10 分, 共 20 分)

1、返回被删除结点的个数

```
Int DeleteNodes(List L , ElmType e) -----0.5 分
{
    List p,q; -----0.5 分
    int number=0; -----0.5 分
    p=L->next; q = L; -----0.5 分
    while(p!=NULL) -----1 分
    {
        if (p->data == e){ -----1 分
            number++; -----1 分
            q->next = p->next; -----1 分
            p=p->next; -----1 分
        }
        else{
            p = p->next; -----1 分
            q = q->next; -----1 分
        }
    }
    return number; -----1 分
}
```

2、递归返回二叉树 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
}
```