



厦门大学《数据结构》期末试题·答案

考试日期：2009.1 (C)

信息学院自律督导部



一、(本题 16 分) 回答下列问题

- (1) 在数据结构课程中, 数据的逻辑结构, 数据的存储结构及数据的运算之间存在着怎样的关系?
- (2) 若逻辑结构相同但存储结构不同, 则为不同的数据结构。这样的说法对吗? 举例说明之。
- (3) 在给定的逻辑结构及其存储表示上可以定义不同的运算集合, 从而得到不同的数据结构。这样说法对吗? 举例说明之。
- (4) 评价各种不同数据结构的标准是什么?

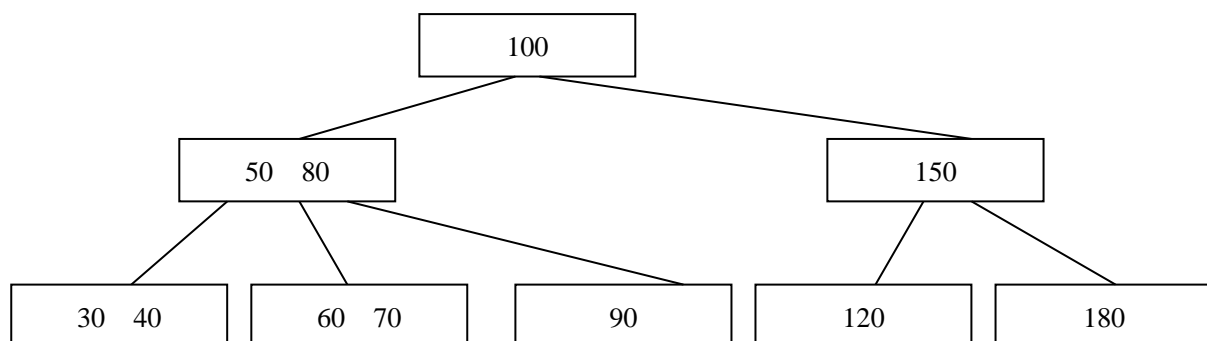
答: (1) 数据的逻辑结构反映数据元素之间的逻辑关系 (即数据元素之间的关联方式或“邻接关系”), 数据的存储结构是数据结构在计算机中的表示, 包括数据元素的表示及其关系的表示。数据的运算是对于数据定义的一组操作, 运算是定义在逻辑结构上的, 和存储结构无关, 而运算的实现则是依赖于存储结构。

(2) 逻辑结构相同但存储不同, 可以是不同的数据结构。例如, 线性表的逻辑结构属于线性结构, 采用顺序存储结构为顺序表, 而采用链式存储结构称为线性链表。

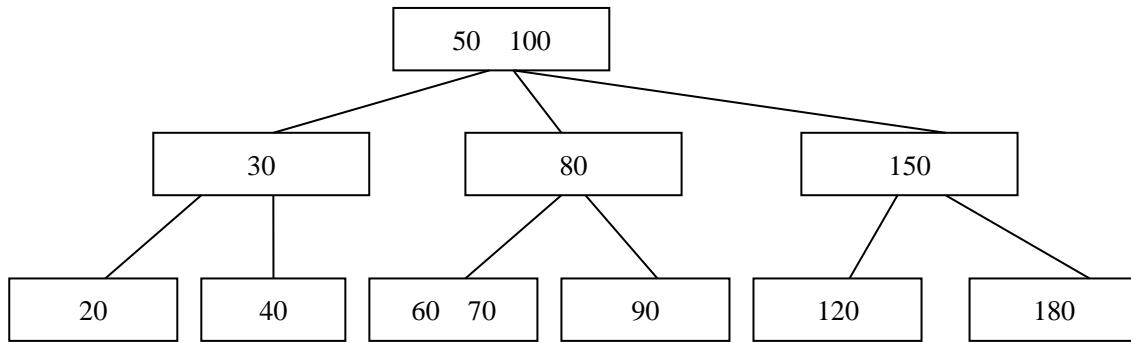
(3) 栈和队列的逻辑结构相同, 其存储表示也可相同 (顺序存储和链式存储), 但由于其运算集合不同而成为不同的数据结构。

(4) 数据结构的评价非常复杂, 可以考虑两个方面, 一是所选数据结构是否准确、完整的刻画了问题的基本特征; 二是是否容易实现 (如对数据分解是否恰当; 逻辑结构的选择是否适合于运算的功能, 是否有利于运算的实现; 基本运算的选择是否恰当。); 三是是否符合待解决问题运行的软硬件环境需要 (如空间、时间等)。

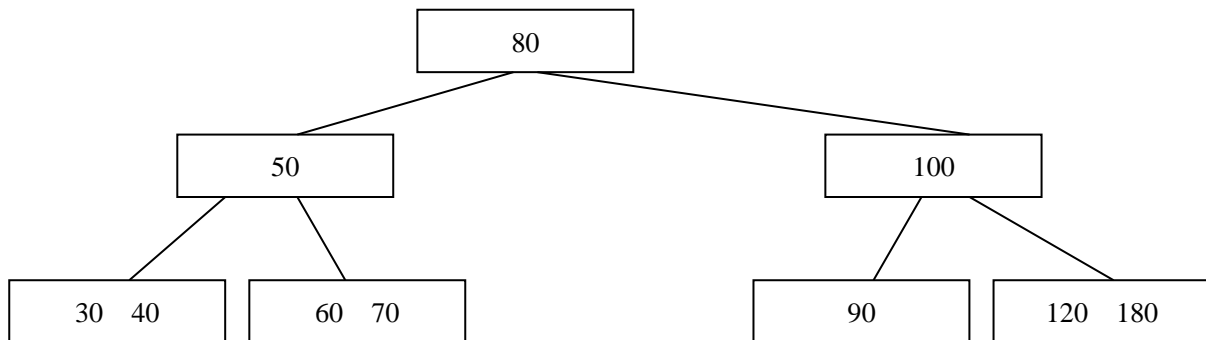
二、(本题 10 分) 设有 3 阶 B-树, 如下图所示, 分别画出在该树插入关键字 20 和在原树删除关键字 150 得到的 B-树。



解: 插入 20 后的 B-树为:



删除 150 后的 B-树为：

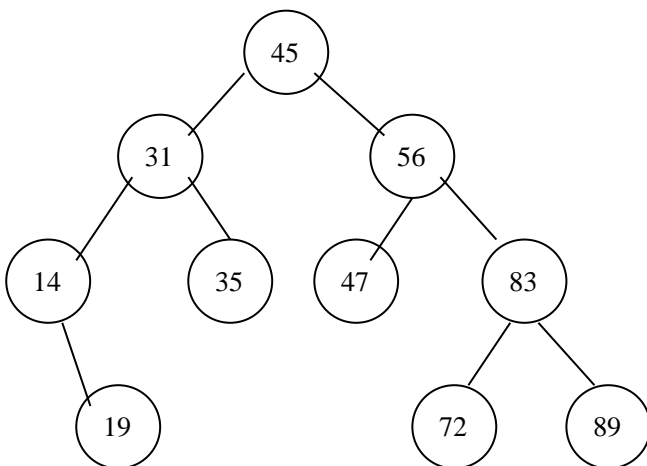


三、（本题 15 分）已知键值序列为 {45, 56, 83, 31, 72, 35, 14, 47, 89, 19}，要求给出：

- （1） 按键值排列次序构造一棵二叉排序树。
- （2） 在等概率的情况下，该二叉排序树查找成功的平均查找长度。
- （3） 针对上述 10 个键值，在不同的排列次序下所构造出的不同形态的二叉排序树中，在最坏和最好情况下，二叉排序树的高度各是多少？

解：总分为 15 分，每一小步 5 分。

（1）



(2) 在等概率情况下, 该二叉排序树的平均检索长度是:

$$ASL = (1 + 2 \times 2 + 3 \times 4 + 4 \times 3) / 10 = 29 / 10 = 2.9$$

(3) 对于上述 10 个键值, 在最坏情况下, 每个结点 (除了叶子结点) 只有右孩子 (或者只有左孩子), 高度为 10。在最好情况下, 高度为 $\lceil \log_2 10 \rceil + 1 = 4$ 。

四、(本题 14 分) 对关键字序列: 49, 38, 66, 80, 70, 15, 22 进行从小到大的排序,

(1) 写出用快速排序法的第一趟和第二趟排序之后的序列状态, 假设选取待排序的第一个关键字为枢轴。

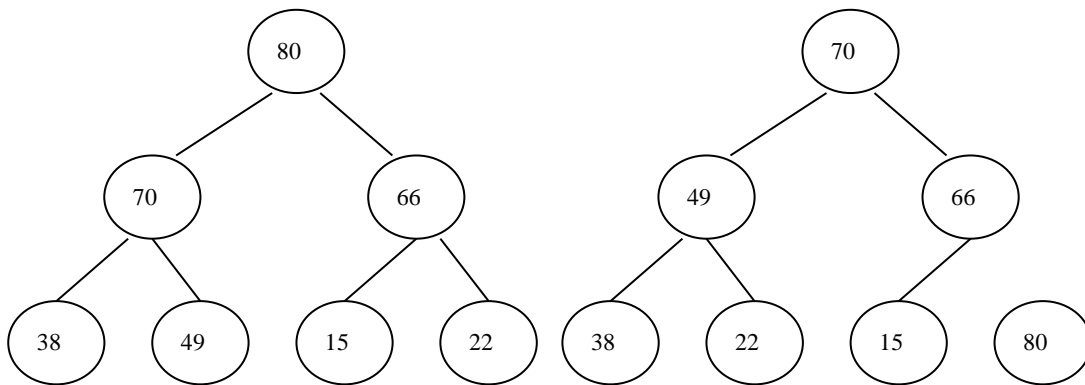
(2) 写出堆排序的初始化构造好的堆以及第一趟和第二趟排序之后的调整好的堆。

答:

(1) 第一趟排序后的状态: (22 38 15) 49 (70 80 66)

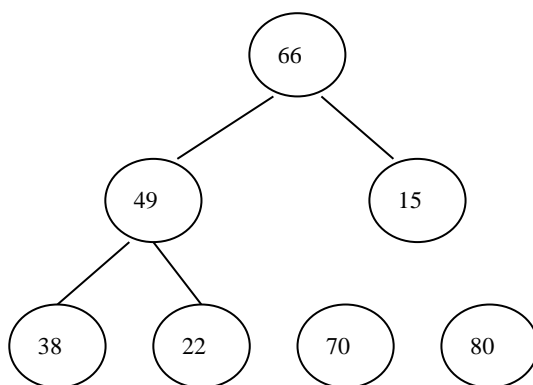
第二趟排序后的状态: (15) 22 (38) (66) 70 (80)

(2)



初始化构造好的堆

第一趟排序之后调整好的堆



第二趟排序之后调整好的堆

五、(本题 15 分) 给出一系列整数, 设计算法求出总和最大的子序列, 要求算法的时间复杂性在 $O(n)$ 之内。

答：

```
int maxsubsum(int a[n])
{
    int maxsum=0; thissum=0; int j=0;
    for (j=0; j<n;j++)
    {
        thissum=thissum+a[j];
        if (thissum>maxsum) maxsum=thissum;
        else if (thissum<0) thissum=0;
    }
    return maxsum;
}
```

时间复杂性为 $O(n)$ 。

六、（本题 10 分）在两个有序线性表中，寻找是否存在共同元素。如果存在共同元素，返回第一个共同元素在第一个有序表中的位置。请设计数据结构，并在其上设计算法。

答：可以参考有序表的归并算法。

数据结构可以使用一维数组，并且第 0 个元素放空。

int SearchCommonItem(int a[n], int b[m])//第 0 位放空，返回值为 0 代表找不到

```
{
    int i=1,j=1;
    while (i<=n && j<=m)
    {
        if (a[i]==b[j]) return i;
        else if (a[i]<b[j]) i++;
        else j++;
    }
    return 0;
}
```

七、（本题 10 分）用孩子兄弟链表作为树的存储结构，设计算法求出树的深度。

解：算法思路：一棵树的深度可以递归定义为：若树为空，则深度为 0，否则树的深度为根结点的所有子树深度的最大值加 1。

数据结构为：

```
typedef struct CSNode{
    ElemType data;
    struct CSNode *firstchild, *nextsibling;
} CSNode, *CSTree;
```

算法如下：

```
int depth(CSNode * t)
{

```

```

CSNode *p; int m, d;
if (t==NULL) return 0;
p=t→firstchild; m=0;
while (p) {
    d=depth(p);
    if (d>m) m=d;
    p=p→nextsibling;
}
return m+1;
}

```

八、（本题 10 分）对于书本内的某一经典算法，提出自己对该算法思想的理解，指出算法的不足之处，并提出自己的改进算法。

答：无标准答案。