

数据结构第8章习题





8-4

- 设有关键词为A、B、C和D，按照不同的输入顺序，共可能组成多少种不同的二叉查找树。请画出其中高度较小的6种。



参考答案

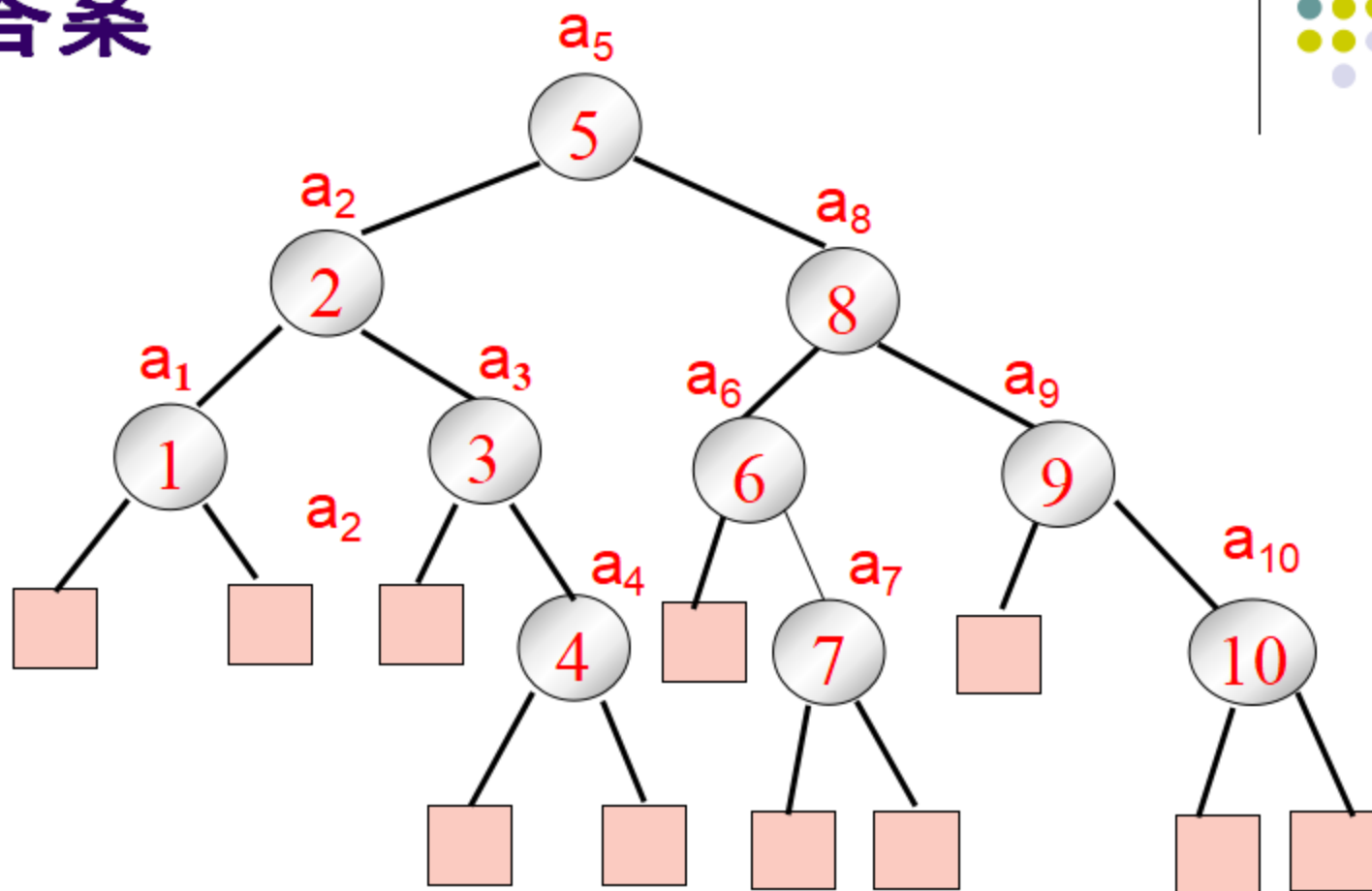
- 以A为根的BST共5种
 - B为第2个元素：2种
 - C为第2个元素：1种（高度为2）
 - D为第2个元素：2种
- 以B为根的BST共2种（高度为2）
- 以C为根的BST共2种（高度为2）
- 以D为根的BST共5种（类似A）
- 一共有14种。高度为2的有6种，为3的有8种



8-7

- 画出对长度为10的有序表进行折半查找的判定树，并求其等概率时查找成功的平均查找长度。
 -

参考答案



- $ASL_{succ} = (1 + 2 \cdot 2 + 3 \cdot 4 + 4 \cdot 3) / 10 = 29 / 10$
- $ASL_{unsucc} = (5 \cdot 3 + 6 \cdot 4) / 11 = 39 / 11$

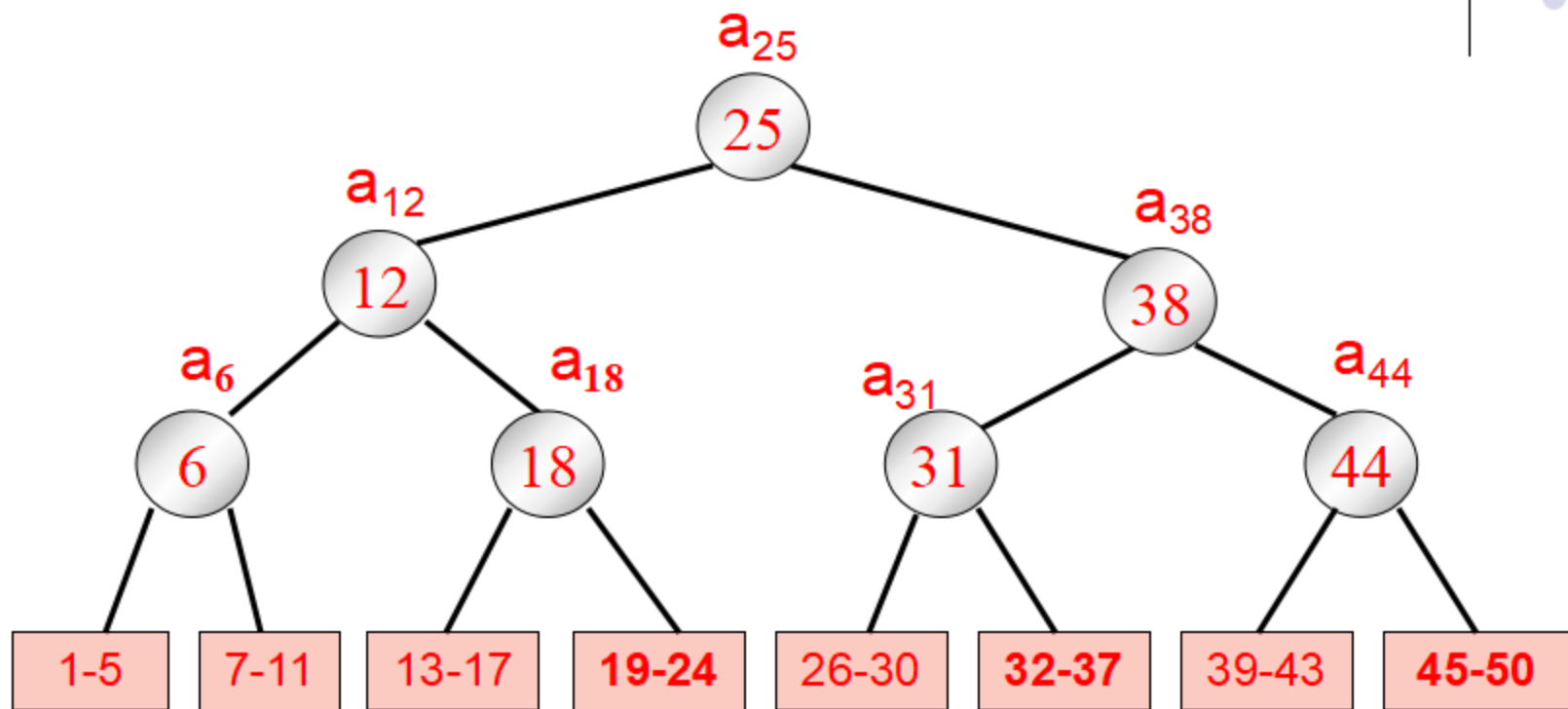


8-9

- 假设按下述递归方法进行顺序表的查找：若表长 $n \leq 10$ ，则进行顺序查找，否则进行折半查找。试画出对表长 $n = 50$ 的顺序表进行上述查找时，描述该查找的判定树，并求出在等概率情况下查找成功的平均查找长度。



参考答案



$$ASL = (1 + 2 \times 2 + 3 \times 4 + (4 + 5 + 6 + 7 + 8) \times 8 + 9 \times 3) / 50 = 284 / 50$$



8-10

- 已知下列关键词和它们对应的散列函数值：

key	Zhao	Sun	Li	Wang	Chen	Liu	Zhang
$H(key)$	6	5	7	4	1	6	4

- 由此构造哈希表，用线性探测法冲突，计算平均查找长度ASL。若用拉链法解决冲突情况又如何？



参考答案

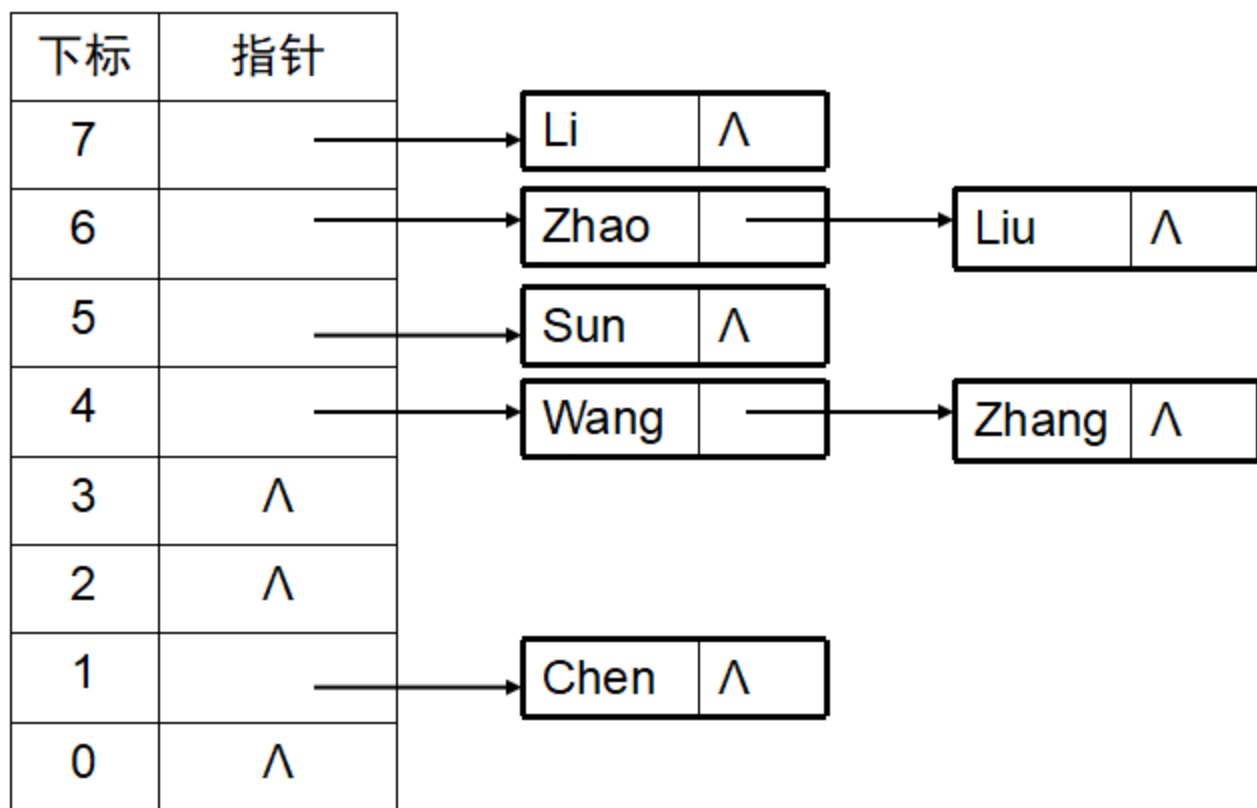
- 线性探测法（假设散列表的长度是8，下标从0开始）

下标	元素
7	Li
6	Zhao
5	Sun
4	Wang
3	
2	Zhang
1	Chen
0	Liu

- 查找成功ASL
$$=(1*5+3*1+7*1)/7$$
$$=15/7$$



- 拉链法（假设散列表长度是8，下标从0开始）



- 查找成功 $ASL = (1 \times 5 + 2 \times 1 + 2 \times 1) / 7 = 9/7$



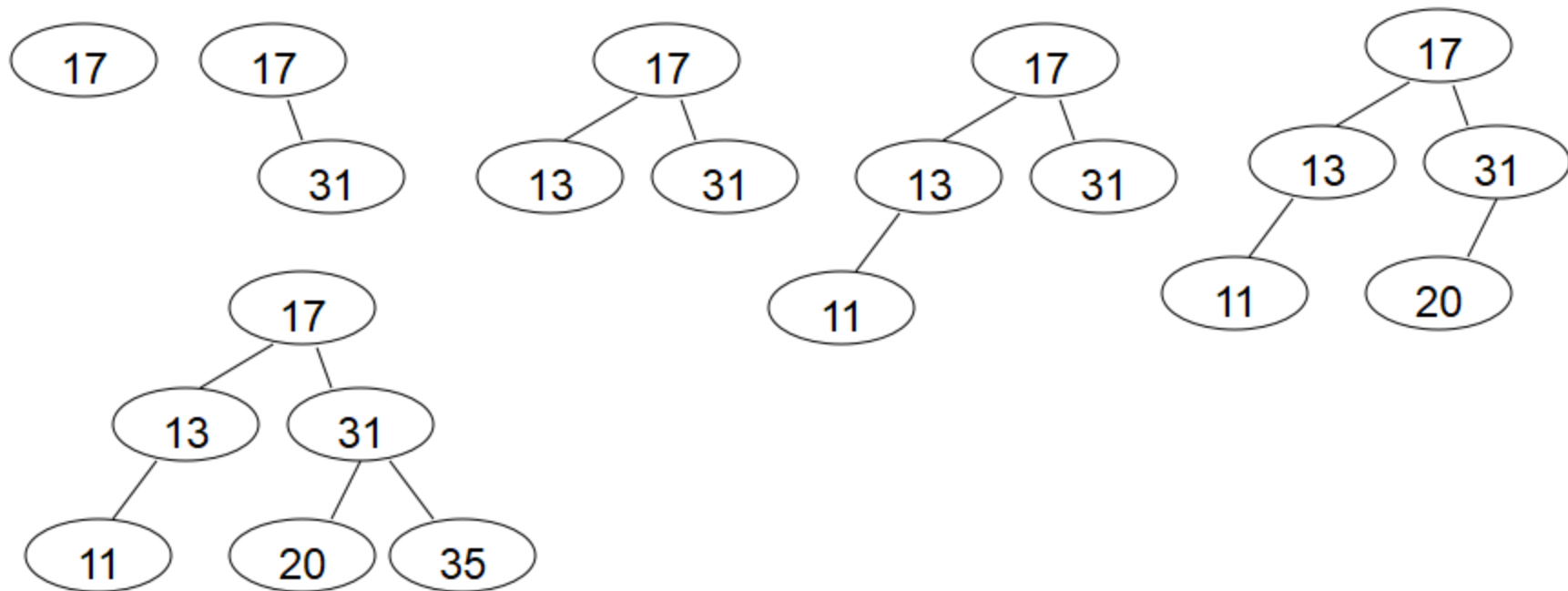
8-13

- 已知序列(17, 31, 13, 11, 20, 35, 25, 8, 4, 11, 24, 40, 27), 请画出该序列的二叉查找树, 并分别给出下列操作后的二叉查找树。
 - (1) 插入数据9
 - (2) 删除节点17
 - (3) 再删除节点13



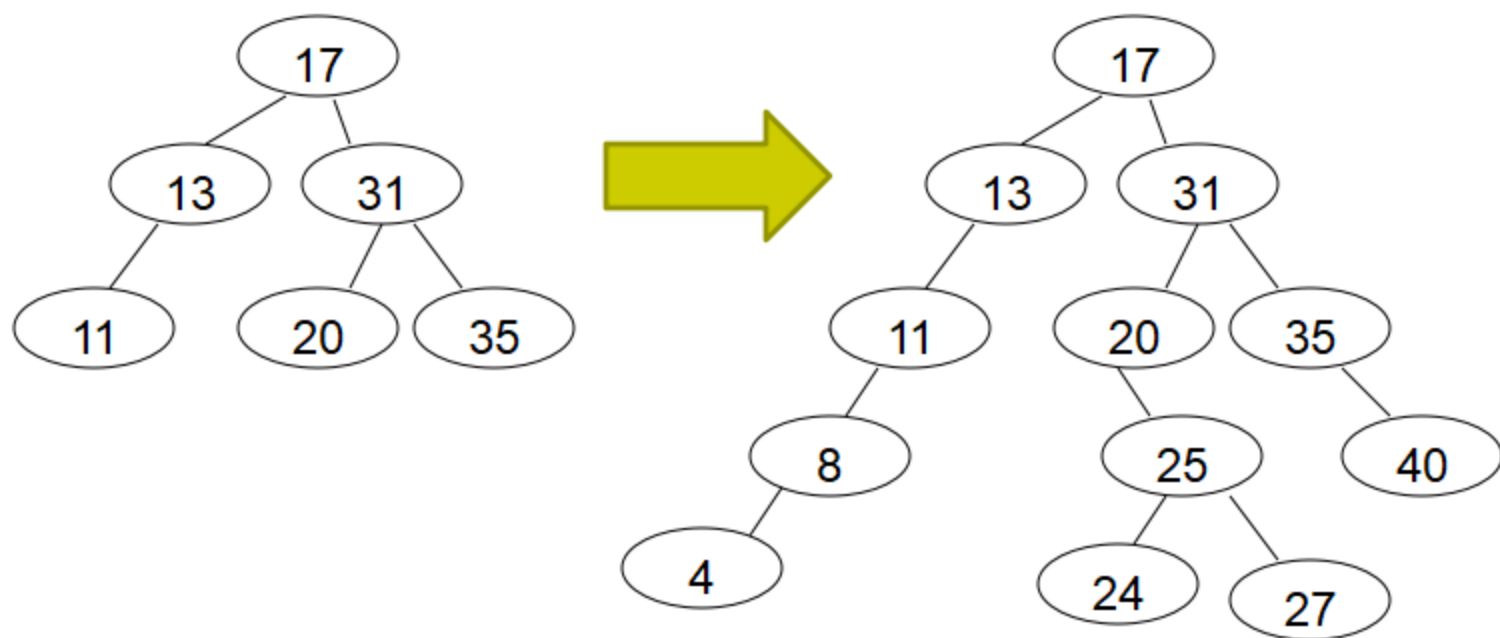
参考答案

- 动态插入创建二叉查找树：
(17, 31, 13, 11, 20, 35, 25, 8, 4, 11, 24, 40, 27)



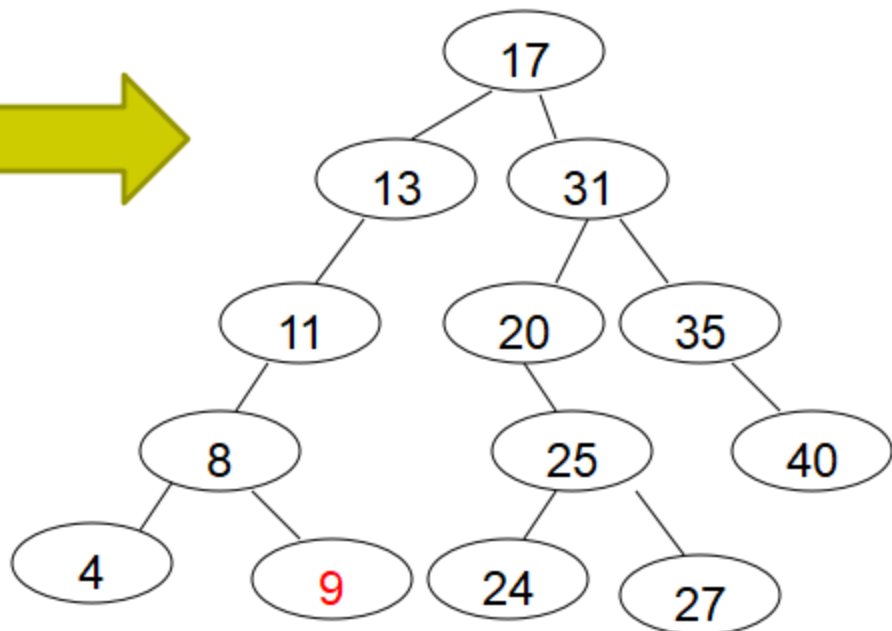
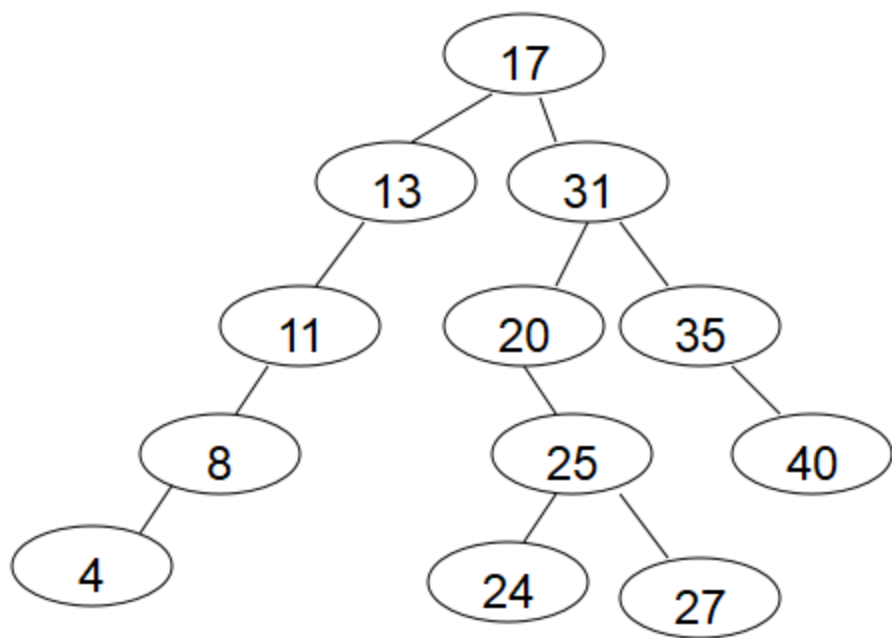


- (17,31,13,11,20,35,25,8,4,11,24,40,27)



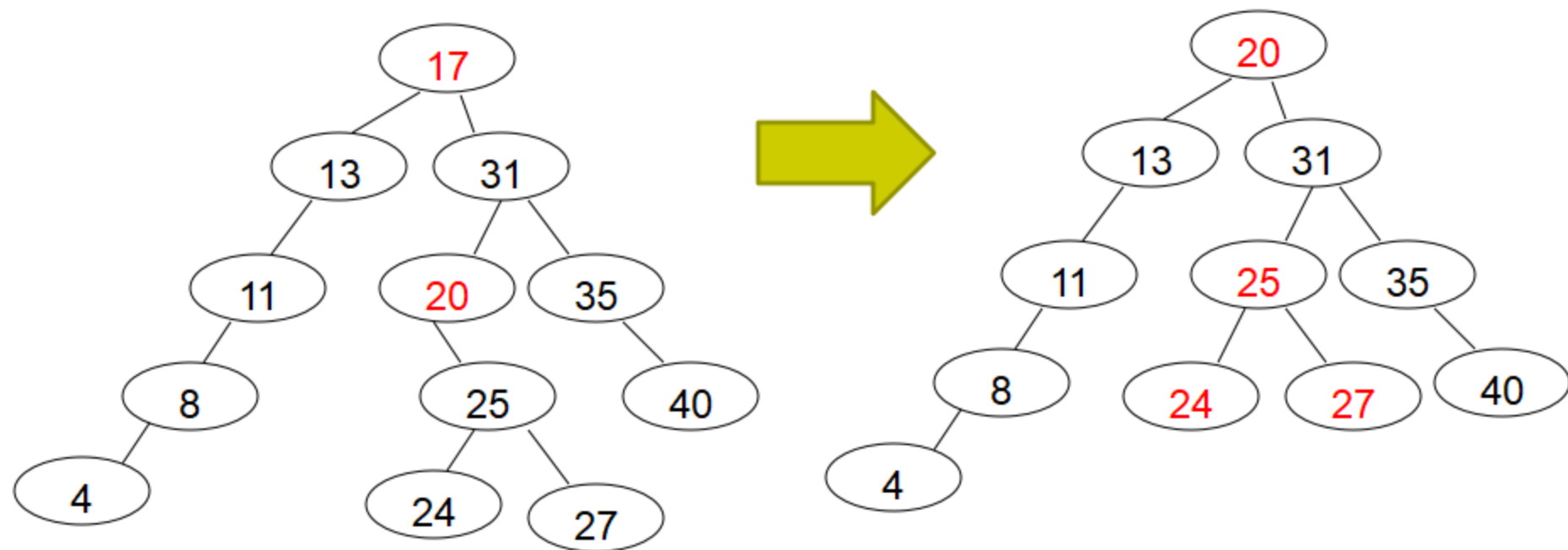


- 插入数据9



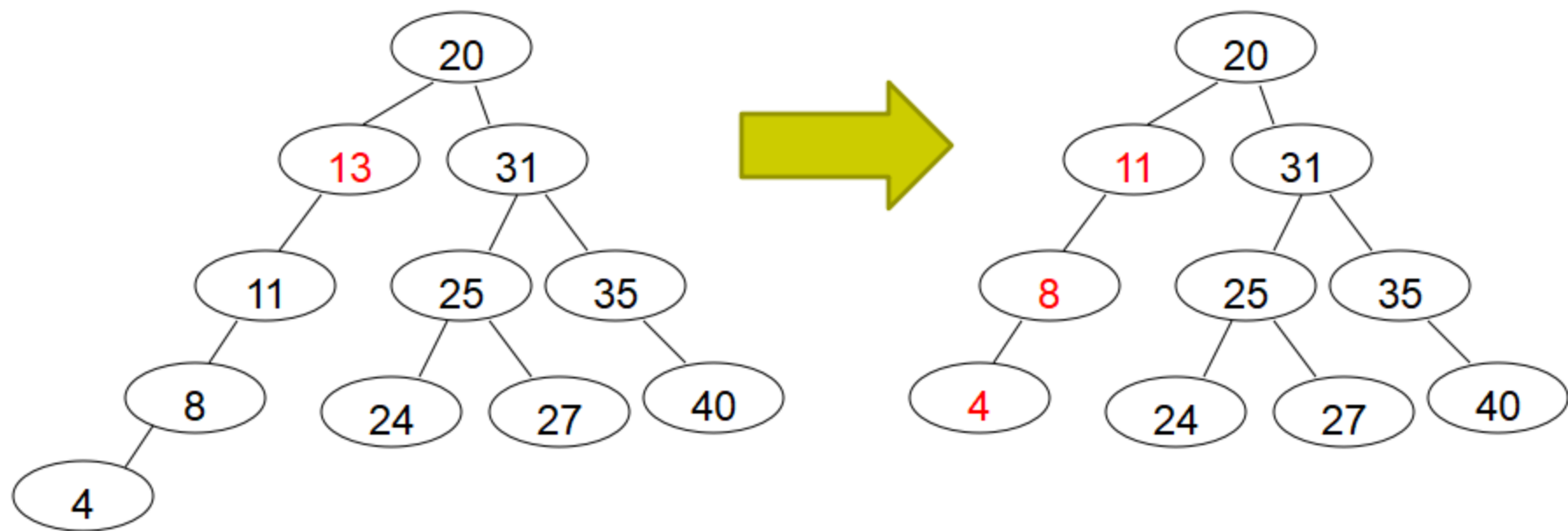


- 删除17





- 再删除节点13



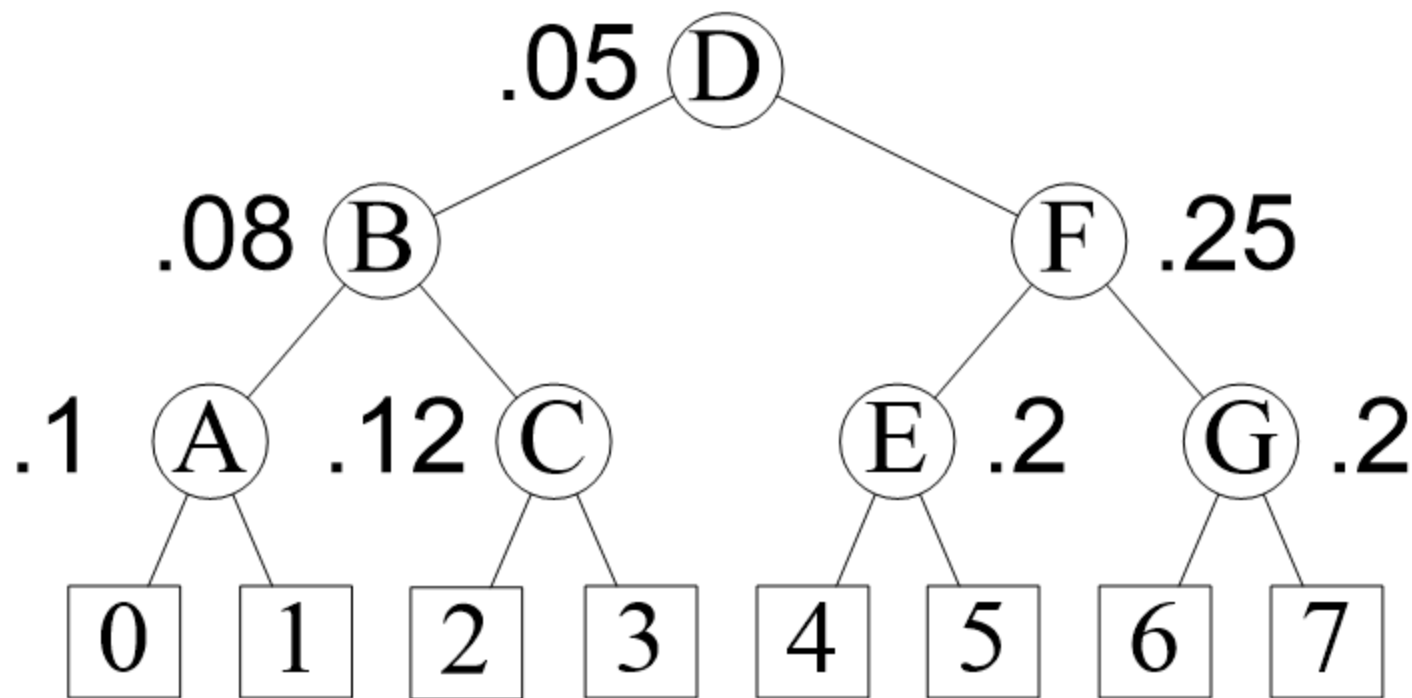


8-14

- 设有一组数据black、blue、green、purple、red、white、yellow，它们的查找概率分别是0.10、0.08、0.12、0.05、0.20、0.25、0.20，试以它们的查找概率为权值，构造一棵次优查找树，并计算其查找成功的平均查找长度。



参考答案（平衡树）



查找成功ASL = $1 \times 0.05 + 2 \times (0.08 + 0.25) + 3 \times (0.1 + 0.12 + 0.2 + 0.2) = 1.86$



8-22

- 编写判定给定的二叉树是否是二叉查找树的算法。



分析

- 判定二叉树是否为二叉查找树是建立在中根遍历的框架基础下，在遍历中附设一指针pre指向当前访问节点的中根直接前驱，每访问一个节点便比较前驱节点pre和此节点是否有序，若遍历结束后各节点和其中根直接前驱均满足有序，则此二叉树为二叉查找树；否则，只要有一个节点不满足，那么此二叉树就不是二叉查找树。



参考答案

算法BST (t, pre, post, flag) /* 使用中根遍历和pre、post指针判断t是否是二叉查找树*/

IF left(t) = NULL && pre != NULL **THEN** post \leftarrow pre.

ELSE (

 BST(left(t), pre, post, flag).

IF !flag **THEN RETURN.**

)

IF post != NULL AND data(post) \geq data(t) **THEN**

 flag \leftarrow FALSE.

ELSE IF right(t) = NULL **THEN** post \leftarrow t.

ELSE BST(right(t), t, post, flag). ■



8-23

- 给定整型数组 $B[0..m, 0..n]$. 已知 B 中数据在每一维方向上都按从小到大的次序排列。且整型变量 x 在 B 中存在。试设计一个算法，找出一对满足 $B[i,j] = x$ 的 i,j 值，要求比较次数不超过 $m+n$.



分析

- 本题中主要是要确定每次进行比较的对象。其中，二维数组右上角的元素是一个较为特殊的元素。可以逐次跟二维数组右上角的元素进行比较。每次比较有3种可能的结果：若相等则结束比较；若右上角的元素小于 x ，则可断定二维数组的最上面一行中肯定不包含 x ，下次比较时搜索范围即可减少一行；若右上角的元素大于 x ，则可断定二维数组的最右面一列中肯定不包含 x ，下次比较时搜索范围即可减少一列。这样，每次至少可使搜索范围减少一列或一行，最多经过 $m+n$ 次即可找到 x 。



参考答案

算法Find(B, m, n, x. i, j)

/*在B中查找x, 时间复杂度 $O(m+n)$ */

F1[初始化]

$i \leftarrow 1. j \leftarrow n.$

F2[比较B[i,j]与x]

```
WHILE  $i \leq m$  AND  $j \geq 1$  DO (  
    IF  $B[i,j] = x$  THEN RETURN.  
    IF  $B[i,j] < x$  THEN  $i \leftarrow i + 1.$   
    IF  $B[i,j] > x$  THEN  $j \leftarrow j - 1.$   
)
```




THE END