



厦门大学《数据结构》期末试题·答案

考试日期：2009·1 (B)

信息学院自律督导部



一、(15 分) n 个不同的整数按递增序列存储在 $T[1..n]$ 中，若存在一个下标 i ，使得 $T[i]=i$ ，写一算法找到该下标，要求算法的最坏时间复杂度为 $O(\log 2n)$ 。

解：本题为 C 类题，总分为 20 分。

使用折半查找法。当存在 $T[i]>i$ 时，应在 $[1..i-1]$ 中查找。

```
int find(int T[], int n)
{
    int low, high, mid;
    low=1; high=n;
    while (low<=high)
    {
        mid=(low+high)/2;
        if (T[mid]==mid) return mid;
        else if (T[mid]>mid) high=mid-1;
        else low=mid+1;
    }
    return 0;
}
```

二、(15 分) 如果一棵度为 m 的树有 n_0 个度为 0 的点， n_1 个度为 1 的点，有 n_2 个度为 2 的点，……，

n_m 个度为 m 的点，试证明：
$$n_0 = \left(\sum_{i=2}^m (i-1)n_i \right) + 1。$$

证明：总结点数 $n = n_0 + n_1 + \dots + n_m$

由于除根结点外，每一个结点都有一个分支进入，因此有

$$b = n - 1 = n_0 + n_1 + \dots + n_m - 1 \quad (1)$$

另外，这些分支都是由度大于 0 的结点所射出的，因此有

$$b = n_1 + n_2 * 2 + \dots + n_m * m。 \quad (2)$$

联立 (1) (2) 式得
$$n_0 = \left(\sum_{i=2}^m (i-1)n_i \right) + 1。$$

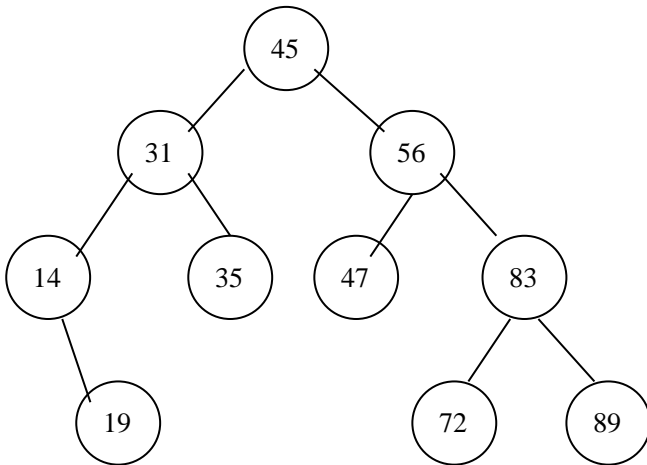
三、(15 分) 已知键值序列为 {45, 56, 83, 31, 72, 35, 14, 47, 89, 19}，要求给出：

(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、(15 分) 荷兰国旗问题：设有一个仅由红、白、蓝三种颜色的条块组成的条块序列。请编写一个时间复杂度为 $O(n)$ 的算法，使得这些条块按红、白、蓝的顺序排好，即排成荷兰国旗图案。

解：程序如下

```

typedef enum {RED, WHITE, BLUE} color; //三种颜色
void Flag_Arrange(color a[ ], int n) //把由三种颜色组成的序列重排为按照红, 白, 蓝的顺序排列
{
    i=0; j=0; k=n-1;

    while(j<=k)

        switch(a[j])

        {

```

```

case RED:

    a[i]<->a[j];

    i++;

    j++;

    break;

case WHITE:

    j++;

    break;

case BLUE:

    a[j]<->a[k];

    k--; //这里没有 j++;语句是为了防止交换后 a[j] 仍为蓝色的情况

}

```

五、（本题 15 分）找出所有满足下列条件的二叉树：

- （1）它们在先序遍历和中序遍历时，得到的遍历序列相同；
- （2）它们在后序遍历和中序遍历时，得到的遍历序列相同；
- （3）它们在先序遍历和后序遍历时，得到的遍历序列相同。

答：

- （1）先序序列和中序序列相同的二叉树为：空树或者任一结点均无左孩子的非空二叉树；
- （2）中序序列和后序序列相同的二叉树为：空树或者任一结点均无右孩子的非空二叉树；
- （3）先序序列和后序序列相同的二叉树为：空树或仅有一个结点的二叉树。