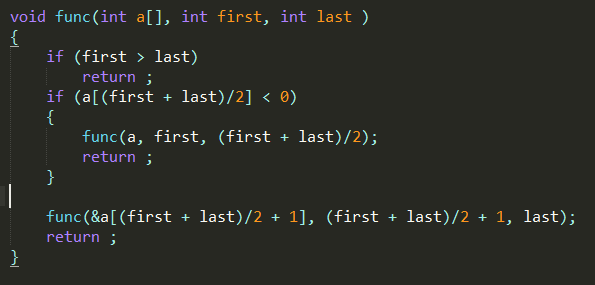
**数据结构2017试题**

**By 赵楠**

1. **一个函数具有下列形式，判断递归调用是否正确，如果不正确，应该如何修改？分析该函数的平均时间复杂度，以O的方式给出，给出分析过程。**



1. **将数组 {32，145，632，88，55，25，135，132，65，13，165，335} 调整为最大堆并使用堆排序将其排为升序数组。要求用图形和数组表示过程。**
2. **编写C语言函数 int is\_index(int a[], int n) 判断升序排列的整数数组a（长度为n）中是否存在a[i] = i；如果存在，返回TRUE，否则返回FALSE。说明程序的时间复杂度是多少。**

题解：

本题只需使用一个for循环遍历数组是否存在满足 a[i] == i条件的即可

具体代码见：<https://github.com/tofar/data-structure/tree/master/数据结构复习卷>

1. **对下列整数进行基数排序，{32，145，632，88，55，25，135，132，65，13，165，335}，要求标出所使用的基以及每一趟桶式排序中的元素变化过程。**
2. **用分治算法的思想计算3192的十位数是多少？说明算法的步骤，给出计算过程。**
3. **利用分治算法的思想编写C语言函数int find\_median(int a[], int m, int b[], int n)，在长度分别为m和n的两个排序整型数组中找出总体的中位值，其中a为升序排列，b为降序排列。说明时间复杂度。中位值定义为第**

**（m+n）/ 2 小的元素。**

**本题参照2016年的第六题即可，只是在开始的时候计算一，取k = （m+n）/ 2，其他一模一样**

具体代码见：<https://github.com/tofar/data-structure/tree/master/数据结构复习卷>

1. **已知一棵二叉树的先序遍历结果为ABCDEFGHI，中序遍历结果为CEDBAGFHI，结点由字母表示。这棵树是否存在？如果存在，请构造出这棵树并标出构造过程。如果不存在，请说明为什么。**
2. **一组符号Si，i = 1 .. 12，其出现的频率分别是3，14，6， 8，15，25，35，12，65，13，33和5。请设计出相应的Huffman编码。要求画出Huffman树，并给出编码。**
3. **编写一个C语言函数int count\_nodes(BinTreeNode \*root)返回以root为根的二叉树中节点的个数。**

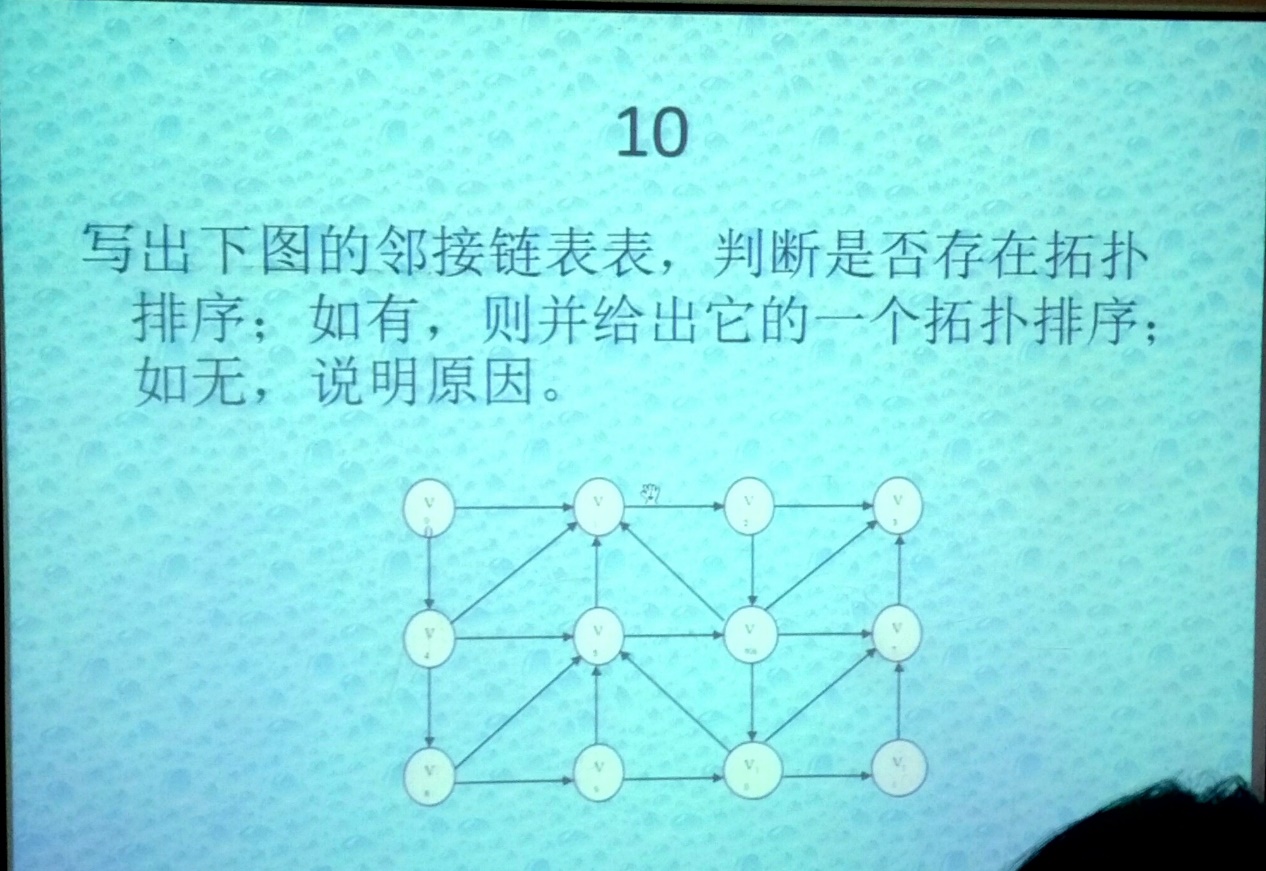
题解：

本题只需使用一个递归返回左子树和右子树的节点数再加 1即可

*count\_nodes(root->left) + 1 + count\_nodes(root->right)*

具体代码见：<https://github.com/tofar/data-structure/tree/master/数据结构复习卷>

1. **写出下图的邻接链表表，判断是否存在拓扑排序；如果存在，则给出它的一个拓扑排序；否则，说明原因。**

****

题解：

参照课本的217页的拓扑排序

注：由于本人看不清那个顺序，这里直接按照从左到右从上到下标号，即

1 2 3 4

5 6 7 8

9 10 11 12

本题不存在一个拓扑排序，由于途中v2->v3->v7->v2构成一个圈，故不存在