

# 牛客网算法基础提升班

---

## 第五课

牛客网最新算法课--基础提升班：针对有简单入门基础，还达不到校招水平要求的同学，左神手把手讲解高级数据结构和算法原理，结合典型例题深入讲解解题思路，并提供最优解和代码

上课时间：每周六日 09:00——11:00

上课老师：左程云，华科本科，芝加哥大学硕士，曾就职于IBM、百度、GrowingIO、亚马逊，也是牛客网的老师。

牛客网：一个提供海量校招真题及专项练习题，笔经面经，招聘信息，学习资源及交流的平台<https://www.nowcoder.com/>



笔经面经



学习交流

## Morris遍历

一种遍历二叉树的方式，并且时间复杂度 $O(N)$ ，额外空间复杂度 $O(1)$

通过利用原树中大量空闲指针的方式，达到节省空间的目的

### Morris遍历细节

假设来到当前节点`cur`，开始时`cur`来到头节点位置

- 1) 如果`cur`没有左孩子，`cur`向右移动(`cur = cur.right`)
- 2) 如果`cur`有左孩子，找到左子树上最右的节点`mostRight`:
  - a. 如果`mostRight`的右指针指向空，让其指向`cur`，然后`cur`向左移动(`cur = cur.left`)
  - b. 如果`mostRight`的右指针指向`cur`，让其指向`null`，然后`cur`向右移动(`cur = cur.right`)
- 3) `cur`为空时遍历停止

## Morris遍历的实质

建立一种机制，对于没有左子树的节点只到达一次，对于有左子树的节点会到达两次

morris遍历时间复杂度的证明

先序、中序可以由morris遍历加工得到

后序遍历也可由morris遍历加工得到，但是把处理时机放在，能够达到两次的节点并且是第二次到达的时候

morris遍历的应用

### 树形dp套路

树形dp套路使用前提：

如果题目求解目标是S规则，则求解流程可以定成以每一个节点为头节点的子树在S规则下的每一个答案，并且最终答案一定在其中

树形dp套路第一步：

以某个节点X为头节点的子树中，分析答案有哪些可能性，并且这种分析是以X的左子树、X的右子树和X整棵树的角度来考虑可能性的

树形dp套路第二步：

根据第一步的可能性分析，列出所有需要的信息

树形dp套路第三步：

合并第二步的信息，对左树和右树提出同样的要求，并写出信息结构

树形dp套路第四步：

设计递归函数，递归函数是处理以X为头节点的情况下的答案。

包括设计递归的**basecase**，默认直接得到左树和右树的所有信息，以及把可能性做整合，并且要返回第三步的信息结构这四个小步骤



### 叉树节点间的最大距离问题

从二叉树的节点**a**出发，可以向上或者向下走，但沿途的节点只能经过一次，到达节点**b**时路径上的节点个数叫作**a到b**的距离，那么二叉树任何两个节点之间都有距离，求整棵树上的最大距离。

## 题目二

派对的最大快乐值

员工信息的定义如下:

```
class Employee {  
    public int happy; // 这名员工可以带来的快乐值  
    List<Employee> subordinates; // 这名员工有哪些直接下级  
}
```

公司的每个员工都符合 **Employee** 类的描述。整个公司的人员结构可以看作是一棵标准的、没有环的多叉树。树的头节点是公司唯一的老板。除老板之外的每个员工都有唯一的直接上级。叶节点是没有任何下属的基层员工(**subordinates**列表为空), 除基层员工外, 每个员工都有一个或多个直接下级。

这个公司现在要办party, 你可以决定哪些员工来, 哪些员工不来。但是要遵循如下规则。

- 1.如果某个员工来了, 那么这个员工的所有直接下级都不能来
- 2.派对的整体快乐值是所有到场员工快乐值的累加
- 3.你的目标是让派对的整体快乐值尽量大

给定一棵多叉树的头节点**boss**, 请返回派对的最大快乐值。

## 提升项目经验

- 课程名称：《牛客高级项目课--（牛客网）》
- 课程地址：<https://www.nowcoder.com/courses/semester/senior>
- 独家内部100元优惠券：DRMscjy



## 面试算法书籍

- 书名：《程序员代码面试指南—IT名企算法与数据结构题目最优解》
- 作者：左程云

# THANK YOU

查看更多笔经面经

