

# 二叉树与分治法

## Binary Tree & Divide Conquer

课程版本 v3.4    主讲 令狐冲



扫描二维码关注微信/微博  
获取最新面试题及权威解答

微信: [ninechapter](#)

微博: <http://www.weibo.com/ninechapter>

知乎: <http://zhuanlan.zhihu.com/jiuzhang>

官网: <http://www.jiuzhang.com>

- 二叉树的深度优先搜索 DFS in Binary Tree
  - 遍历问题 Preorder / Inorder / Postorder
  - 分治算法 Introduce Divide Conquer Algorithm
  - 非递归 遍历法 分治法 Non-recursion vs Traverse vs Divide Conquer
  - 二叉搜索树 Binary Search Tree
    - Insert / Remove / Find / Validate
- 二叉树的宽度优先搜索 BFS in Binary Tree

# Time Complexity Training

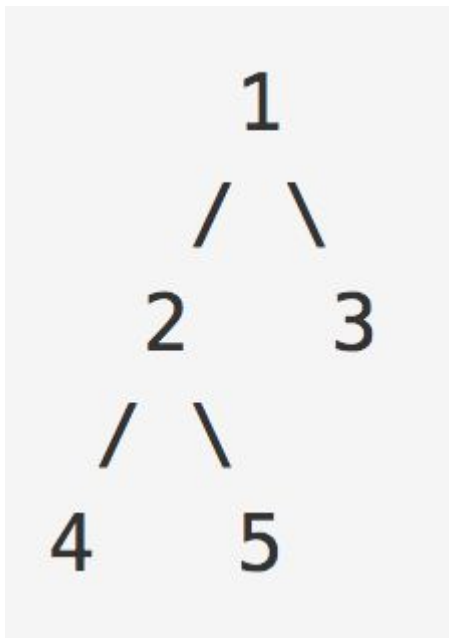
通过 $O(n)$ 的时间，把 $n$ 的问题，变为了 $n/2$ 的问题，复杂度是多少？

通过 $O(1)$ 的时间，把 $n$ 的问题，变成了两个 $n/2$ 的问题，复杂度是多少？



# Preorder Postorder Inorder

- Preorder 前序遍历
  - 1 245 3 根左右
- Inorder 中序遍历
  - 425 1 3 左根右
- Postorder 后序遍历
  - 452 3 1 左右根



- Preorder:
  - <http://www.lintcode.com/problem/binary-tree-preorder-traversal/>
  - <http://www.jiuzhang.com/solutions/binary-tree-preorder-traversal/>
- Inorder
  - <http://www.lintcode.com/en/problem/binary-tree-inorder-traversal/>
  - <http://www.jiuzhang.com/solutions/binary-tree-inorder-traversal/>
- Postorder:
  - <http://www.lintcode.com/en/problem/binary-tree-postorder-traversal/>
  - <http://www.jiuzhang.com/solutions/binary-tree-postorder-traversal/>

# Divide Conquer Algorithm

- Traverse vs Divide Conquer
  - They are both Recursion Algorithm
  - Result in parameter vs Result in return value
  - Top down vs Bottom up
- Merge Sort / Quick Sort
- 90% Binary Tree Problems!

DFS 深度优先搜索

用递归实现

用非递归实现

遍历法

分治法

# 独孤九剑 —— 破枪式

碰到二叉树的问题，想想整棵树在该问题上的结果  
和左右儿子在该问题上的结果之间的联系是什么

# 令狐大师兄带你写程序之

<http://www.lintcode.com/problem/maximum-depth-of-binary-tree/>

<http://www.lintcode.com/problem/minimum-depth-of-binary-tree/>



# Balanced Binary Tree

<http://www.lintcode.com/problem/balanced-binary-tree/>

<http://www.jiuzhang.com/solutions/balanced-binary-tree/>

When we need ResultType?

# Lowest Common Ancestor

<http://www.lintcode.com/problem/lowest-common-ancestor/>

<http://www.jiuzhang.com/solutions/lowest-common-ancestor/>

with parent pointer vs no parent pointer

# Take a break

5 分钟后回来

# Binary Tree Maximum Path Sum II

<http://www.lintcode.com/problem/binary-tree-maximum-path-sum-ii/>

<http://www.jiuzhang.com/solutions/binary-tree-maximum-path-sum-ii/>

Root to Any

# Binary Tree Maximum Path Sum

<http://www.lintcode.com/problem/binary-tree-maximum-path-sum/>

<http://www.jiuzhang.com/solutions/binary-tree-maximum-path-sum/>

Any to Any

会做这道题，才敢说自己会分治

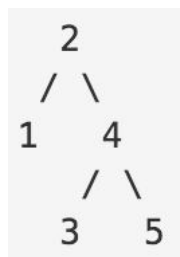
# Binary Search Tree

二叉查找树，简称“BST”

又名“二叉搜索树”“排序二叉树”

- 从定义出发：
  - 左子树都比根节点小
  - 右子树都比根节点大
    - 如果存在重复元素，可以自行选择放到左子树还是右子树

- 从效果出发：
  - 中序遍历 in-order traversal 是升序序列
  - 如图，中序遍历为 1 2 3 4 5



- 性质：
  - 如果一棵二叉树的中序遍历不是升序，则一定不是BST
  - 如果一棵二叉树的中序遍历是升序，也未必是BST
    - 当存在重复元素时，相同的数要么同时在左子树，要么同时在右子树，不能一边一个

# Validate Binary Search Tree

<http://www.lintcode.com/problem/validate-binary-search-tree/>

<http://www.jiuzhang.com/solutions/validate-binary-search-tree/>

traverse vs divide conquer



# Binary Search Tree Iterator

<http://www.lintcode.com/en/problem/binary-search-tree-iterator/>

<http://www.jiuzhang.com/solutions/binary-search-tree-iterator/>

Iterator vs Inorder with non-recursion

## Related Questions

---

- In-order Successor in Binary Search Tree
- <http://www.lintcode.com/problem/inorder-successor-in-binary-search-tree/>
- <http://www.jiuzhang.com/solutions/inorder-successor-in-binary-search-tree/>
- Search Range in Binary Search Tree
- <http://www.lintcode.com/problem/search-range-in-binary-search-tree/>
- Insert Node in a Binary Search Tree
- <http://www.lintcode.com/problem/insert-node-in-a-binary-search-tree/>
- Remove Node in a Binary Search Tree
- <http://www.lintcode.com/problem/remove-node-in-binary-search-tree/>
- <http://www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/9-BinTree/BST-delete.html>

# BFS in Binary Tree

BFS = Breadth First Search

二叉树上的宽度优先搜索算法

# 令狐大师兄带你写程序之

<http://www.lintcode.com/problem/binary-tree-level-order-traversal/>

<http://www.jiuzhang.com/solutions/binary-tree-level-order-traversal/>

# Binary Tree Level Order Traversal

- 2 Queues
- 1 Queue + Dummy Node
- **1 Queue (Best)**
- Follow up: Can you do it in DFS?

## Related Questions

---

- Binary Tree Level Order Traversal II
- <http://www.lintcode.com/problem/binary-tree-level-order-traversal-ii/>
- <http://www.jiuzhang.com/solutions/binary-tree-level-order-traversal-ii/>
- Binary Tree Zigzag Level Order Traversal
- <http://www.lintcode.com/problem/binary-tree-zigzag-level-order-traversal/>
- <http://www.jiuzhang.com/solutions/binary-tree-zigzag-level-order-traversal/>

# 今天学会了什么？

- 用树形分析法计算时间复杂度
- 递归是深度优先搜索算法(DFS)的一种实现形式
  - DFS可以使用非递归的方式实现
- 二叉树上的递归 Recursion in Binary Tree
  - 遍历法 Traverse
  - 分治法 Divide Conquer
- 二叉搜索树
  - 性质: 中序遍历是升序序列
  - 功能:  $O(h)$ 的时间查找, 删除, 插入
- 二叉树上的宽度优先搜索
  - 使用一个队列的宽度优先搜索算法
    - 在二叉树上无需使用 Hash 判重
  - 如何实现分层遍历
- 必“背”程序:
  - 非递归版本的 Pre Order, In Order
  - 二叉树分层遍历

# 点题时间

<http://www.jiuzhang.com/qa/983/>