

树和堆

Homebrew的作者因为不能手写翻转二叉树被Google拒了...

七月在线 林应

2018年3月

主要内容

- ☐ 树的定义
- ☐ 树的应用
- ☐ 常用算法
- ☐ 实战
- ☐ 堆的定义
- ☐ 堆的应用
- ☐ 实战
- ☐ 作业



树的定义 (by Wiki)

- 树（英语：Tree）是一种无向图（undirected graph），其中任意两个顶点间存在唯一一条路径。或者说，只要没有回路的连通图就是树。
- 二叉树（英语：Binary tree）是每个节点最多只有两个分支（不存在分支度大于2的节点）的树结构。通常分支被称作“左子树”和“右子树”。二叉树的分支具有左右次序，不能颠倒。



树的定义 (by Wiki)

- ❑ 完全二叉树：叶节点只能出现在最下层和次下层，并且最下面一层的结点都集中在该层最左边的若干位置的二叉树。
- ❑ 平衡二叉树：它是一棵空树或它的左右两个子树的高度差的绝对值不超过1，并且左右两个子树都是一棵平衡二叉树。



树的应用

□ 快速数据检索

- STL的红黑树

- 数据库的B+树

□ 文档结构组织：DOM

□ 人工智能：决策树

□ 游戏：通过构造空间树实现快速碰撞检测

(<https://www.zhihu.com/question/2511128>)



树的应用

■ 区块链的默克尔树

- 为什么要检验交易的一致性？
- 100万条，甚至1000万条记录中有1到2条被篡改，如何快速找到被修改的记录？
- 如何利用二分搜索的特性来做hash比较？



常用算法

□ 递归

■ 树的深度优先遍历（以前序遍历为例）

□ 根节点，左子树，右子树

■ 左子树：根节点，左子树，右子树

■ ...

■ 右子树：根节点，左子树，右子树

■ ...

□ 模拟遍历如下二叉树（详情见图1.jpg）

■ 二叉查找树

常用算法

□ 队列

■ 树的广度优先遍历（分层遍历）

- 两个队列，一个队列保存当前要遍历的节点，另一个队列用于存储下次遍历用的节点。初始状态当前队列只有一个元素根节点。
- 然后弹出工作队列元素，如果有子节点，推入下次遍历用节点。当当前队列为空，切换2个队列的功能继续。
- 详情见图2.jpg

实战

- ❑ 二叉树的前中后序遍历（递归）
- ❑ 翻转二叉树（递归）
- ❑ 非递归翻转二叉树（队列）
- ❑ 二叉树的锯齿形层次遍历（队列）
- ❑ 中序遍历和后序遍历树构造二叉树（递归）
- ❑ 非递归中序遍历（堆栈）



堆的定义（from Wiki）

- 堆的实现通过构造二叉堆（binary heap），实为二叉树的一种；由于其应用的普遍性，当不加以限时，均指该数据结构的这种实现。这种数据结构具有以下性质：
 - 任意节点小于（或大于）它的所有后裔，最小元（或最大元）在堆的根上（堆序性）。
 - 堆总是一棵完全树。即除了最底层，其他层的节点都被元素填满，且最底层尽可能地从左到右填入。

堆的定义（from Wiki）

□ 建堆的过程

■ 详情见“图3.jpg”

堆的定义（from Wiki）

□ 建堆的复杂度分析

- N 个节点的堆高度最大为 $h = \log N$ ，最下面一层非叶子节点最多调整1次，倒数第2层最多2次，…依此类推，根节点最多需要 h 次。
- 最下面一层子节点共有 $2^{(h-1)}$ 个，倒数第2层有 $2^{(h-2)}$ 个，…依此类推，根节点有 $2^{(h-h)}$ 个1个。
- 所以总的时间复杂度为 $1^{(h-1)} + 2 * 2^{(h-2)} + (h-1) * 2 + h$ ，得到结果为 $N * 2 - 2 - \log(N)$ ，所以时间复杂度 $O(n)$ 。

堆的应用

□ 优先任务调度

- 操作系统如何根据线程优先级进行快速调度？

□ 海量数据多路归并排序

- 面试最爱问题，海量数据排序选择topXXX。

实战

- 给定一个数组，生成一个最小堆。
- 在上题的基础上，实现合并k个排序链表。

作业

□ 非递归前序遍历二叉树

(<https://leetcode.com/problems/binary-tree-preorder-traversal/>)

□ 排序矩阵中的从小到大第k个数

(<https://leetcode.com/problems/merge-k-sorted-lists/>)

- 在一个排序矩阵中找从小到大的第 k 个整数。
- 排序矩阵的定义为：每一行递增，每一列也递增。
- 例子，给出k=4，矩阵为[[1, 5, 7], [3, 7, 8], [4, 8, 9]]，返回5。



感谢大家！

恳请大家批评指正！