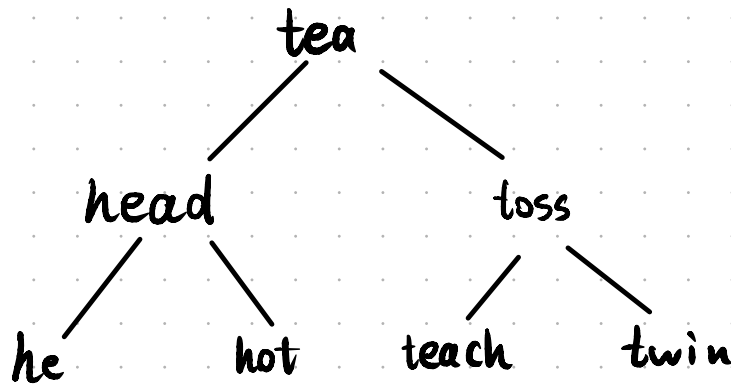


1. 从空二叉树开始, 依次按 AVL 插入算法插入  $\{head, he, tea, teach, twin, hot, toss\}$  (按字典序, 如  $he < tea < teach$ ), 请给出插入后的 AVL 树。



2. 在矩阵中大部分元素均为 0 时, 我们可以采用稀疏矩阵的形式来存储它。假设我们有一个  $N \times N$  阶稀疏矩阵  $A$ , 其中非 0 元素共有  $M$  个。用  $M$  个三元组  $(i, j, A_{ij})$  来表示矩阵  $A$ , 并且按行优先顺序由小到大存储。(即首先比较  $i$ , 若  $i$  一样大则再比较  $j$ )。请给出时间复杂度  $O(M + N)$  的方法, 来将矩阵  $A$  转置。(文字描述、伪代码描述均可)。

首先使用  $n$  个队列:  $q_0, \dots, q_{N-1}$  代表  $N$  列

再遍历  $M$  个元素:

对元素  $(i, j, A_{ij})$ :

① 交换  $i, j$  得到三元组  $(j, i, A_{ij})$

② 将三元组压入队列  $q_j$

再遍历  $N$  个队列:

对每个队列按顺序弹出所有元素并按队列先后排好

得到转置后矩阵的稀疏矩阵

由于  $i$  递增, 从而得到的每个队列中的三元组按列 (转置后) 递增

根据算法过程, 时间复杂度为  $O(M + N)$

3. 在构建二叉搜索树时, 有时会有很多相同的关键字。假设我们要将 $n$ 个相同的关键字依序插入到空的二叉搜索树当中。

(1) 采用标准的构建方法, 若当前节点为 $x$ , 插入关键字 $z$ , 则 $z < x$ 时将 $z$ 插入 $x$ 的左子树; 否则插入右子树。这种情况下插入的总时间复杂度为多少?

(2) 修改插入算法, 在每个树种的节点 $x$ 处设立一个布尔变量 $x.b$ 。插入新关键字 $z$

时, 若 $z = x$ , 则根据 $x.b = \text{True}$  或  $\text{False}$  决定插入左子树还是右子树, 而后令

$x.b \rightarrow !x.b$ 。请给出时间复杂度。

(3) 在每个 $x$ 处设置一个列表存储关键字为 $x$ 的数据。请给出时间复杂度。

(4) 等概率随机选取左子树还是右子树。请给出最坏条件下的时间复杂度及平均条件下的时间复杂度。(平均时间复杂度不用严格证明, 大致说明即可)

(1) 此时总是插入右子树

总比较次数为  $1 + 2 + \dots + n - 1 = \frac{n(n-1)}{2}$

时间复杂度为  $O(n^2)$

(2) 对每个已插入结点, 每个待插入结点遇到它

的时候, 总是将前后两个待插入结点插入不同子树

因此这样总能保证树高最矮, 树高  $O(\log n)$

时间复杂度为  $O(n \log n)$

(3) 此时所有关键字均插入root结点的列表中

插入 $n$ 次, 每次放入列表中 $O(1)$ , 时间复杂度为  $O(n)$

(4) 最坏: 总插入每个已插入结点的一边

与(1)相同,  $O(n^2)$

平均: 对每个结点等概率将待插入结点左右分配:

与(2)相同,  $O(n \log n)$

