AVL Tree

AVL树

描述：

AVL树最早发明的一种自平衡二叉查找树，树中的任何节点的左右两个子树的高度最大差别为1，因此也成为高度平衡树。AVL树的查找、插入、删除操作的平均时间复杂度都是。

为了保持树的左右子树的平衡，避免一边过长或过短，AVL树会对LL（左左）、RR（右右）、LR（左右）、RL（右左）四种情况进行调整：









上面四种情况包含了所有从不平衡转化为平衡的步骤，其中LL和RR，通过单向右旋平衡处理左左

是一种高效的key-value存储结构，时间复杂度为，通过散列函数对key值计算，得到value在哈希表数组中应该存储的位置。散列函数/哈希函数是哈希表的核心，一般形式为，具有速度快的优势，设计好的散列函数的碰撞率低、数据分布平均的优点，可以将任意情况下的数据都分散在哈希表内部的数组中。