AVL树：带有平衡条件的二叉查找树。平衡主要是保证二叉查找树的操作（某些可能的插入操作除外，当插入操作破坏了平衡条件）的时间为。

AVL树的平衡条件：每个结点的左子树和右子树的高度最多差1.

树的高度：根结点到一片树叶的最长路径的长。

空树（没有任何结点，包括根结点的树）的高度定义为-1。主要是为了方便计算只有一个根结点的树的高度为0。



结点的左子树高度：从结点的左子树计算的结点的高度。

结点的右子树高度：从结点的右子树计算的结点的高度。



结点A的左子树高度为3，右子树高度为2。

结点B的左子树高度为2，右子树高度为1.

结点G的左子树高度为0，右子树高度为0.

结点E的左子树高度为0，右子树高度为1.

树的旋转：AVL树执行了插入操作后，有可能会破坏平衡条件，这时需要对树进行修正，以使其能够重新满足平衡条件。

AVL树插入某个结点导致树的平衡性被破坏，只有插入结点到根结点的路径上的结点的平衡条件可能会被破坏。应该在第1个需要重新平衡的结点（最深的结点，可以理解为离根结点最远的结点）处来平衡AVL树，将这个结点记为α结点，α结点的左右子树的高度差肯定为2。α结点的不平衡总共有以下4种可能引起：

1. 对α的左儿子的左子树进行了一次插入（右单旋转重新平衡）。
2. 对α的右儿子的右子树进行了一次插入（左单旋转重新平衡）。
3. 对α的左儿子的右子树进行了一次插入（先左旋后右旋重新平衡）。
4. 对α的右儿子的左子树进行了一次插入（先右旋后左旋重新平衡）。