树的遍历是树的一种重要的运算。所谓遍历是指对树中所有节点的信息的访问，即依次对树中每个节点访问一次且仅访问一次，我们把这种对所有节点的访问称为遍历

树的两种重要的遍历模式是

深度优先遍历

广度优先遍历

深度优先遍历：

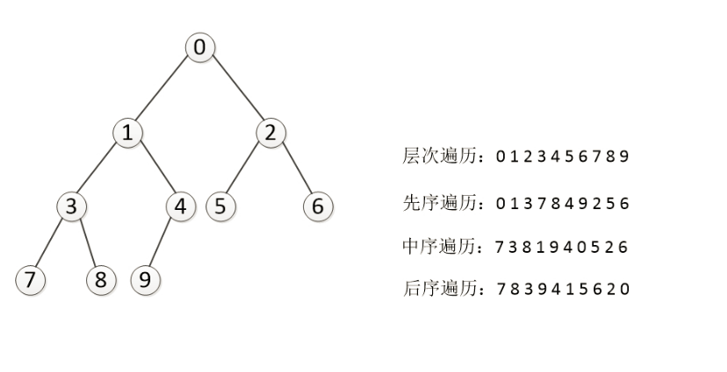
对于一颗二叉树，深度优先搜索是沿着树的深度遍历树的节点，尽可能深得搜索树的分支

有三种方式常被用于访问树的节点，他们之间的不同在于访问每个节点的次序不同：

1、先序遍历 -- 根节点->左子树->右子树

2、中序遍历--左子树->根节点->右子树

3、后序遍历--左子树->右子树->根节点



广度优先遍历

从树的根节点开始，从上到下从左到右遍历整个树的节点

**def** breadth\_travel**(**self**,** root**):**

"""利用队列实现树的层次遍历"""

**if** root **==** **None:**

**return**

queue **=** **[]**

queue**.**append**(**root**)**

**while** queue**:**

node **=** queue**.**pop**(**0**)**

**print** node**.**elem**,**

**if** node**.**lchild **!=** **None:**

queue**.**append**(**node**.**lchild**)**

**if** node**.**rchild **!=** **None:**

queue**.**append**(**node**.**rchild**)**