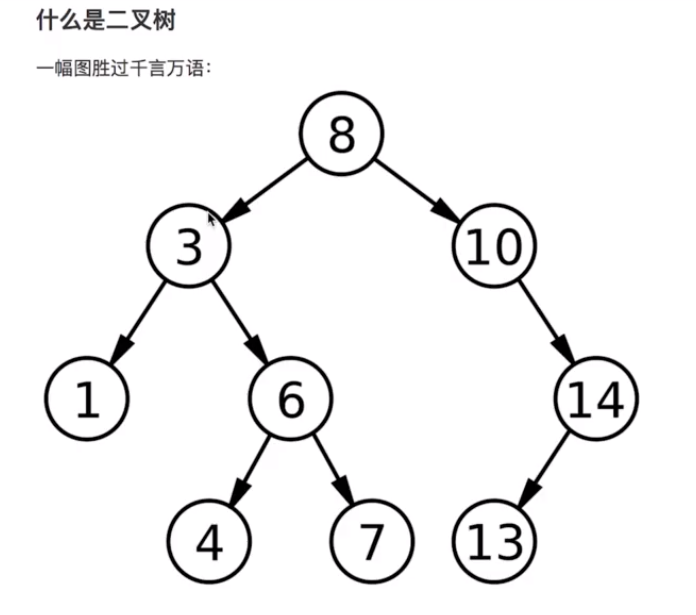
# 什么是二叉树



# 排序二叉树

父亲节点大于做孩子节点，小于右孩子节点。

# 中序遍历

先打印左边节点，再打印当前节点，最后打印右边节点。按照从小到大排序。

# 前序遍历

为了复制一颗二叉树，比重新构造一颗二叉树，效率要高。（在数据量节点大的情况下）

先打印当前节点，再打印左边节点，再打印右边节点。

# 后续遍历

文件系统中的遍历。先看左右子树是否有节点。

先打印左边节点，再打印右边节点，最后打印当前节点。

# 二叉树查找

查找最小值：

查找最大值：

查找指定值：

# 二叉查找树（BST）

具备什么特性呢？其实也就是二叉排序树

1.**左**子树上所有结点的值均**小于或等于**它的根结点的值。

2.**右**子树上所有结点的值均**大于或等于**它的根结点的值。

3.左、右子树也分别为二叉排序树。

二分查找的思想：查找所需的最大次数等同于二叉查找树的高度。

# 红黑树

自平衡的二叉查找树

1.节点是红色或黑色。

2.根节点是黑色。

3.每个叶子节点都是黑色的空节点（NIL节点）。

4 每个红色节点的两个子节点都是黑色。(从每个叶子到根的所有路径上不能有两个连续的红色节点)

5.从任一节点到其每个叶子的所有路径都包含相同数目的黑色节点。

下图中这棵树，就是一颗典型的红黑树：

