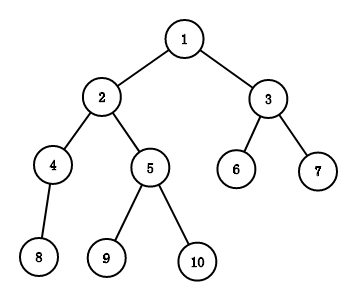
## 概要

对于有根树，LCA代表的是树上两点的最近公共祖先。

什么是祖先呢，若存在一条从根向下的路径，A与B在路径中，且A比B更靠近根，那么A就是B的祖先。

从根开始dfs，可以发现，祖先总是比子孙先遍历到，后遍历出，所以在dfs时，记录每个点的进入时间be和离开时间ed，就能快速地判断两个点的祖孙关系，因为祖先的时间序会将子孙的时间序夹在中间。



继续说LCA，如图，4与10的LCA为2,1与7的LCA为1，发现奇怪的地方了吗？LCA问题里，允许节点自己为自己的祖先。换个角度理解，A与B的LCA为A与B的路径上，深度最小（最靠近根的那个节点）。如果考虑从根到叶的路径，若A、B处于同一条路径，则LCA是深度更小的那个点，若处于不同路径，则LCA为2条路径的分叉点。

那么LCA有什么用呢？一个经典的应用就是快速求树上任意两点间的距离，因为A与B的距离，就是A与B分别与它们的LCA的距离之和。而对于有祖孙关系的两点，距离是非常好求的，记录每个点的深度（或者说是离根的距离），两点的减一下就是了。

## 求法一 转为RMQ问题

并不是什么好的方法，写起来麻烦，不推荐。

先假设原树是一棵二叉树，记录下起中序遍历序列，因为是中序遍历，所以任意两点的公共祖先

## 求法二 倍增

数有其二进制表示，所以一个数可以分解成，其二进制形式上，为1的位对应的位权和。节点向上走N步，等于依次走各个为1的位对应的位权步。