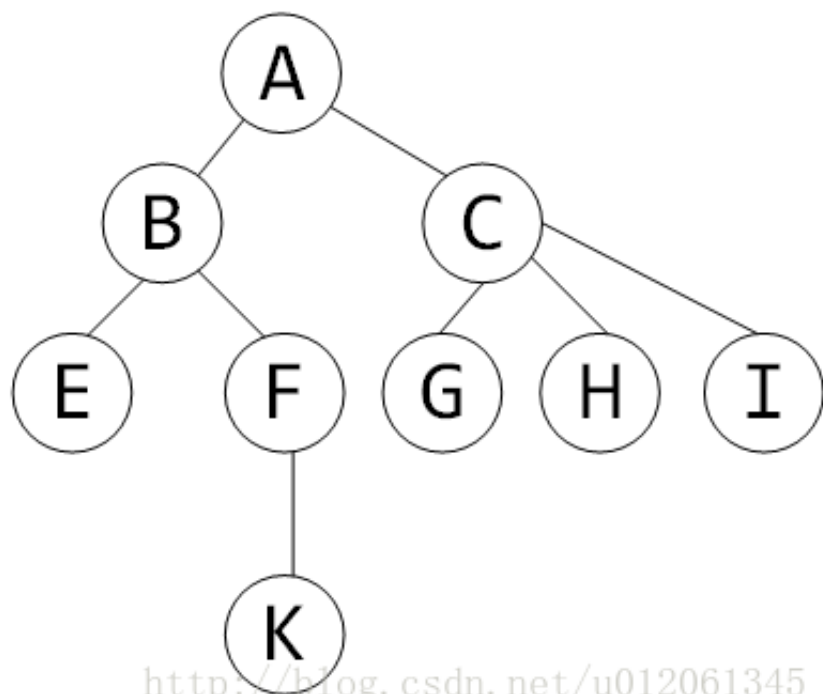
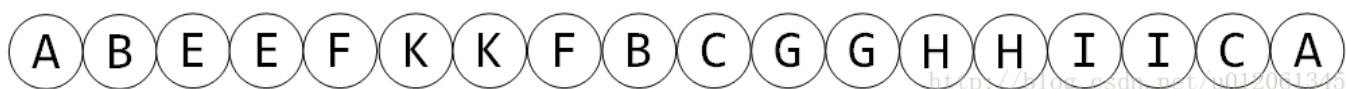


树是一种非线性结构，一般而言，我们总是想办法将其转化为线性结构，将树上操作包括子树操作、路径操作等转化为数组上的区间操作，从而在一个较为理想的复杂度内加以解决。将树“拍平”的方法有很多，例如欧拉序、HLD等。实际上欧拉序也是在DFS过程中得到的。不过通常而言，我们所说的DFS序是指：**每个节点进出栈的时间序列**。



考虑上图中树的DFS序，应为



其中，每个节点均会出现2次，第一次是进入DFS的时刻，第二次是离开DFS的时刻。分别称之为In与Out。在区间操作中，如果某个节点出现了2次，则该节点将被“抵消”。所以通常会将Out时刻对应的点设置为负数。

树的DFS序列有几个有用的性质：

1. 任意子树是连续的。例如子树BEFK，在序列中对应BEEFKKFB；子树CGHI，在序列中对应连续区间CGGHHIIC。
2. 任意点对(a, b)之间的路径，可分为2种情况，首先令lca是a、b的最近公共祖先：
  1. 若lca是a、b之一，则a、b之间的In时刻的区间或者Out时刻区间就是其路径。例如AK之间的路径就对应区间ABEEFK，或者

KFBCGGHHIICA 。

2. 若lca另有其人，则a、b之间的路径为In[a]、Out[b]之间的区间或者In[b]、Out[a]之间的区间。另外，还需额外加上lca考虑EK 路径，对应为EFK 再加上B。考虑EH 之间的路径，对应为EFKKFBCGGH 再加上A。

来源: <https://blog.csdn.net/u012061345/article/details/54023285>