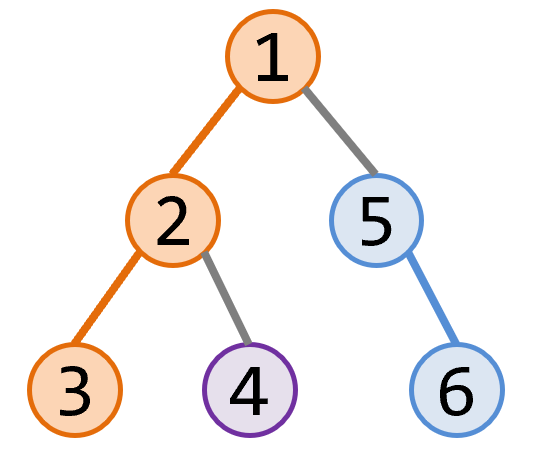
2561.重链剖分1

今天我们来学习树的重链剖分：如何将一棵树剖分成若干链条。对于每个节点u，在u的儿子里，选取子树节点最多的儿子叫做“重儿子”。如果两个儿子对应子树大小相同，我们约定选取编号较小的儿子为“重儿子”。u的其他儿子都称为“轻儿子”。注意：若u只有一个儿子，这个儿子一定就是“重儿子”。因为叶节点没有儿子，所以叶节点没有重儿子。如图：2是重儿子，3是重儿子，4是轻儿子，5是轻儿子，6是重儿子。



每个节点和它“重儿子”之间的连边称作“重边”，每个节点和它“轻儿子”之间的连边称作“轻边”。如图：彩色的是重边，灰色的是轻边。将相邻的“重边”连接起来，就形成了“重链”。注意，另外还有种特殊的“重链”只包含1个节点：如果叶节点v是其他节点的轻儿子时，v就形成了单个节点的特殊“重链”。

每个叶节点一定是某条“重链”的链尾，同时每条“重链”的结尾一定是某个叶节点，所以叶节点的数量就恰好等于重链的数量。

已知一棵树共n个节点，编号1到n，根节点是1号。请输出所有重链的链头节点编号。

输入文件 decomposition.in 输入第一行为正整数n，n<=100000。之后共n-1行，每行为2个正整数u,v表示u和v之间有一条连边。

输出文件decomposition.out 输出一行若干个正整数，由空格隔开，代表重链链头节点的编号，从小到大排列。

输入样例：

6

1 2

2 3

2 4

1 5

5 6

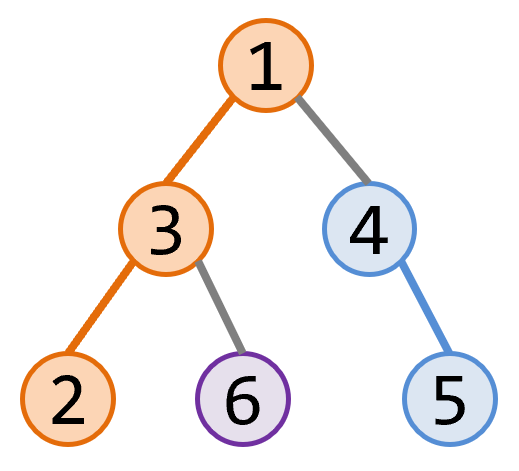
输出样例：

1 4 5

2562. 重链剖分2

今天我们来学习树的重链剖分：如何将一棵树剖分成若干链条。对于每个节点u，在u的儿子里，选取子树节点最多的儿子叫做“重儿子”。如果两个儿子对应子树大小相同，我们约定选取编号较小的儿子为“重儿子”。u的其他儿子都称为“轻儿子”。

每个节点和它“重儿子”之间的连边称作“重边”，每个节点和它“轻儿子”之间的连边称作“轻边”。将相邻的“重边”连接起来，就形成了“重链”。注意，另外还有种特殊的“重链”只包含1个节点：如果叶节点v是其他节点的轻儿子时，v就形成了单个节点的特殊“重链”。



已知一棵树共n个节点，编号1到n，根节点是1号。请对该树进行深度优先搜索(DFS)，搜索时优先选重儿子访问，输出DFS序列以及每个节点的DFN。

输入文件 heavyfirst.in 输入第一行为正整数n，n<=100000。之后共n-1行，每行为2个正整数u,v表示u和v之间有一条连边。

输出文件heavyfirst.out 输出共两行，第一行共n个正整数表示DFS序列，由空格隔开。第二行共n个正整数表示每个节点的DFN，第i个数为i号节点在DFS序列里第几个被访问。注意：行末不可以有空格。

输入样例：

6

1 3

2 3

3 6

1 4

4 5

输出样例：

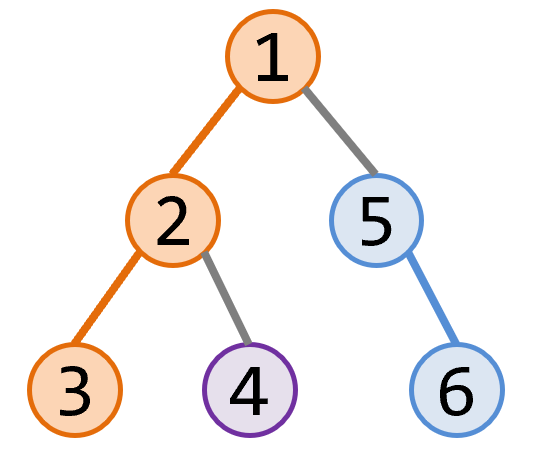
1 3 2 6 4 5

1 3 2 5 6 4

2563. 重链剖分3

今天我们来学习树的重链剖分：如何将一棵树剖分成若干链条。对于每个节点u，在u的儿子里，选取子树节点最多的儿子叫做“重儿子”。如果两个儿子对应子树大小相同，我们约定选取编号较小的儿子为“重儿子”。u的其他儿子都称为“轻儿子”。

每个节点和它“重儿子”之间的连边称作“重边”，每个节点和它“轻儿子”之间的连边称作“轻边”。将相邻的“重边”连接起来，就形成了“重链”。注意，另外还有种特殊的“重链”只包含1个节点：如果叶节点v是其他节点的轻儿子时，v就形成了单个节点的特殊“重链”。



已知一棵树共n个节点，编号1到n，根节点是1号。现在共有m个关于“路径剖分”的问询，每个问询形式为：对于节点x和y，从x到y的路径共经过几条不同的重链。

输入文件 pathdecomposition.in 输入第一行为正整数n，n<=100000。之后共n-1行，每行为2个正整数u,v表示u和v之间有一条连边。接着一行为正整数m，m<=100000。之后共m行，每行为两个正整数x,y表示问询关于x到y的路径。

输出文件pathdecomposition.out 输出一行共m个正整数，由空格隔开，代表答案。

输入样例：

6

1 2

2 3

2 4

1 5

5 6

3

3 6

4 5

1 3

输出样例：

2 3 1

拓展题：

1887. 公司破事之五

​lester是"VVJFJ方熊玥题"公司的老板，共有n名员工，工号分别为1...n（lester本人是1号员工）。公司是一个官僚体系，除lester外每个人有且仅有一个直属上级。为了加强中央集权，lester决定所有事情都需要逐级上报到他亲自决定。每个员工有一个懒惰值（包括lester），早上刚上班时，懒惰值都是0

每次，当一个员工x有事要汇报时，他本人的懒惰值**首先**会增加w（w取决于事情的麻烦程度），然后他会将此事逐级上报直到lester，这个过程所花时间为所有经办人（包括x及其所有上级，含lester）**当前**的懒惰值之和。注意上报的时候，不会增加上级们本身的懒惰值，但其（之前积累的）懒惰值将计入本次操作的时间；计算中，x本人此时的懒惰值是包含当次增加的w。

输入文件company.in 第1行2个正整数n,m。

后n-1行每行2个正整数u,v，表示u,v间是直属上下级关系，但你需要自己判断谁是上级

后m行每行2个正整数x,w，依次表示员工x要汇报一件麻烦程度为w的事。

输入文件company.out 共m行每行1个整数，依次表示每次操作的实际耗时，取模100000007的余数。

输入样例

3 3

1 2

2 3

2 2

3 1

3 2

输出样例

2

3

5

数据规模：

30%数据，1<=n,m<=100

100%数据，1<=n,m<=100000，1<=u,v,x<=n，1<=w<=100000

​