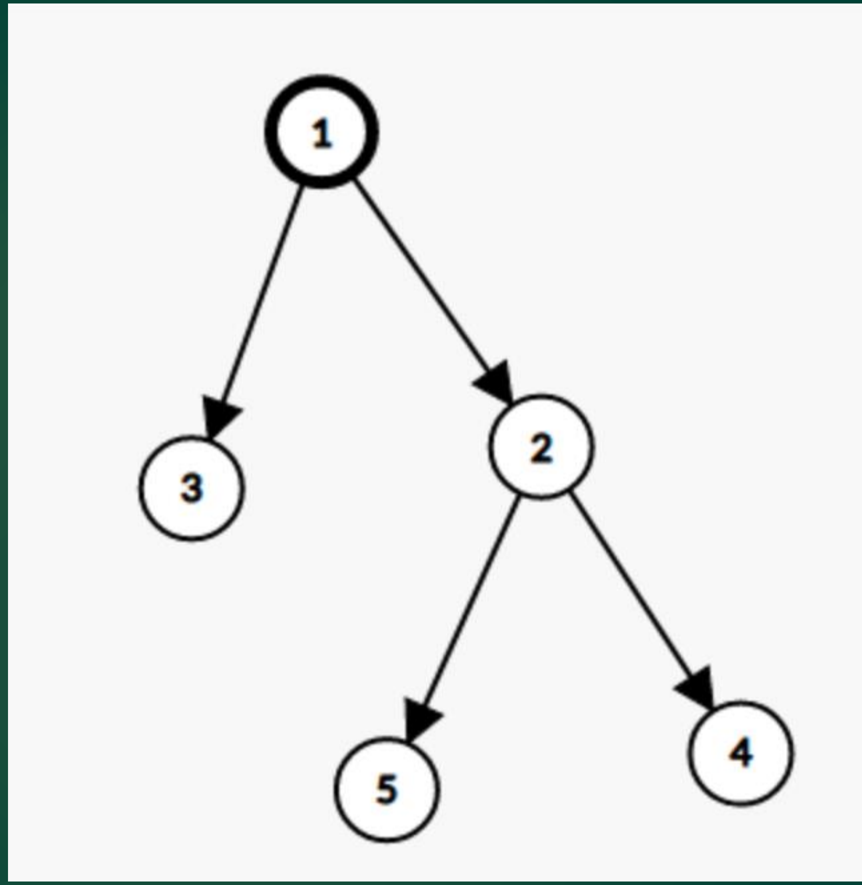
2599最近公共祖先（LCA）

题目描述：

LCA（Lowest Common Ancestors），即最近公共祖先，是指在有根树中，找出某两个节点u和v最近的公共祖先。(若u为v的祖先或者v为u的祖先，则LCA(u,v)就是作为祖先的那个节点)

现在有一棵以1为根的有根树，树上一共有n个节点。现在有m次查询，每次询问两个节点的LCA。



如图所示的数据中，LCA(3,5)=1,LCA(2,5)=2,LCA(3,4)=1,LCA(2,4)=2,LCA(5,5)=5.

2599最近公共祖先（LCA）

输入描述:

第1行：一个正整数n,表示树上节点的个数。(1<=n<=1000)

第2行-第n行：每行两个正整数u,v，表示节点u到节点v有一条边。(1<=u,v<=n)

第n+1行：一个正整数m，表示查询的次数。(1<=m<=1000)

第n+2-第n+m+1行：每行两个正整数a,b，表示要查询的两个节点的编号。(1<=a,b<=n)

输出描述:

对于每个询问，输出一个正整数表示答案，并换行。

样例输入:

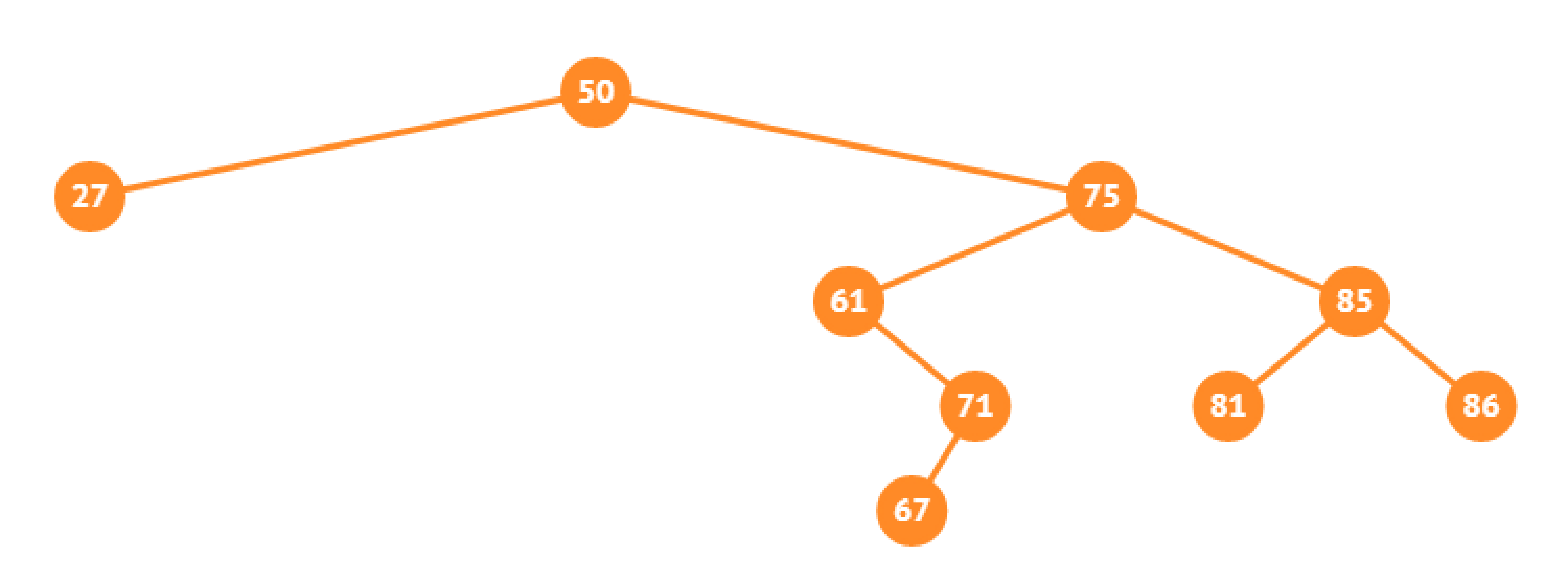
```
5
1 2
1 3
2 4
2 5
5
3 5
2 5
3 4
2 4
5 5
```

样例输出:

```
1
2
1
2
5
```

2599最近公共祖先（LCA）-解题思路

根据本题的数据范围，我们可以采用暴力求解的方式处理每一个查询。



2599最近公共祖先（LCA）-解题思路

暴力求解的思路是这样的，我们可以先分别求出两个节点到根节点的路径。然后分别沿着这两条路径向下走，直到找两条路径上第一个不同的点，那么最后一个相同的点，就是这两个点的LCA。

以上图中81和86为例，81的路径是50,75,85,81，86的路径是50,75,85,86。这两条路径中，最后一个相同的节点是85 ，所以两个点的LCA是85。

这个方法还有一个可以改进的地方，如果处理得当，并不需要遍历整个路径，就能找到两个点的LCA 。这个方法需要做预处理工作，预先知道每个节点的深度。这部分内容我们在之前已经讲过，只需要一次O(N)复杂度的DFS即可。

2599最近公共祖先（LCA）-解题思路

以81和86为例，我们分别从这两个节点向上走，希望他们在某一步，走到相同的节点上，而那个节点就是他们的LCA 。实际上81和86各走一步后，同时到达了85，所以他们的LCA是85。

但上面的方法存在一个问题，如果两个节点深度不相等，你走一步我走一步，可能永远无法走到同一个节点，因为深度较小的已经走到了前面，而后面的永远追赶不上。这时应该如何处理呢？我们可以让深度较大的节点先走，走到与较小的深度一致。

以 86和67为例，86的深度为3， 67的深度为4 。我们先让67走到71 ，此时两个节点的深度相同，再每次各走一步，走到75为止。

除了这种暴力的方法之外，关于LCA的快速求法还有很多，有的可以做到平摊O(1)的复杂度，这些留在以后讲。

2621树上距离

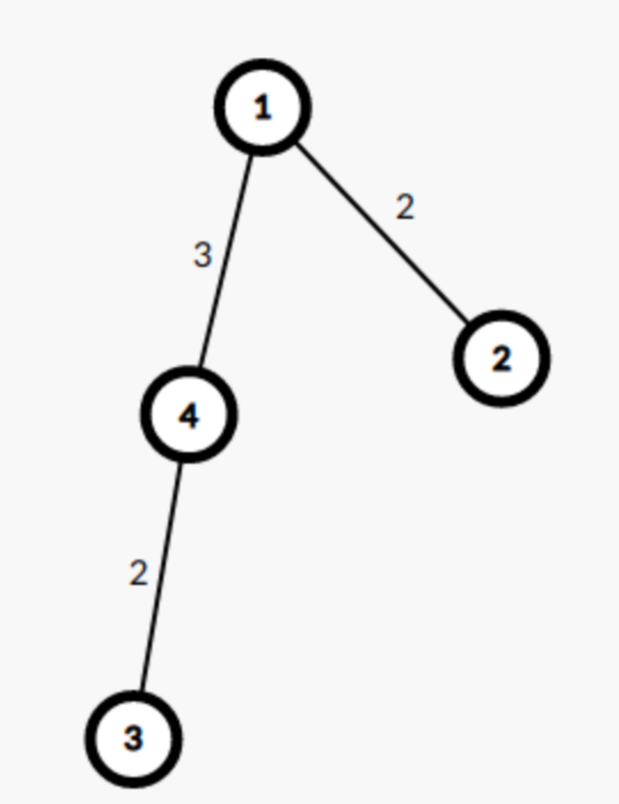
题目描述：

有一棵n个节点的无向树，每条边有一个边权，现在有q次询问，每次询问给出两个点，求这两个点之间的简单路径上的边权和是多少。

如图所示的数据中：

1号节点和2号节点之间的距离为：2

2号节点和3号节点之间的距离为：2+2+3=7



输入描述：

第1行：两个整数n和q,n表示这棵树的节点个数，q表示查询的次数。(1<=n,q<=1000)

第2行~第n行：每行有三个整数u,v,w，表示u与v之间有一条权值为w的边。(1<=w<=10000)

第n+1行~n+q行：每行有两个正整数x,y，表示要查询的两个点的编号。

输出描述：

第1行-第q行：每行输出一个数，表示那要查询的两点之间的简单路径上的边权和。

样例输入:
4 2
2 1 2
4 3 2
1 4 3
1 2
3 2
样例输出:
2
7

2621树上距离-解题思路

如何快速计算路径上的边权和呢？假如我们记录了每个点到根的边权和H_i，则从i到j的路径边权和：

$H_i + H_j - H_{LCA(i,j)} \times 2$

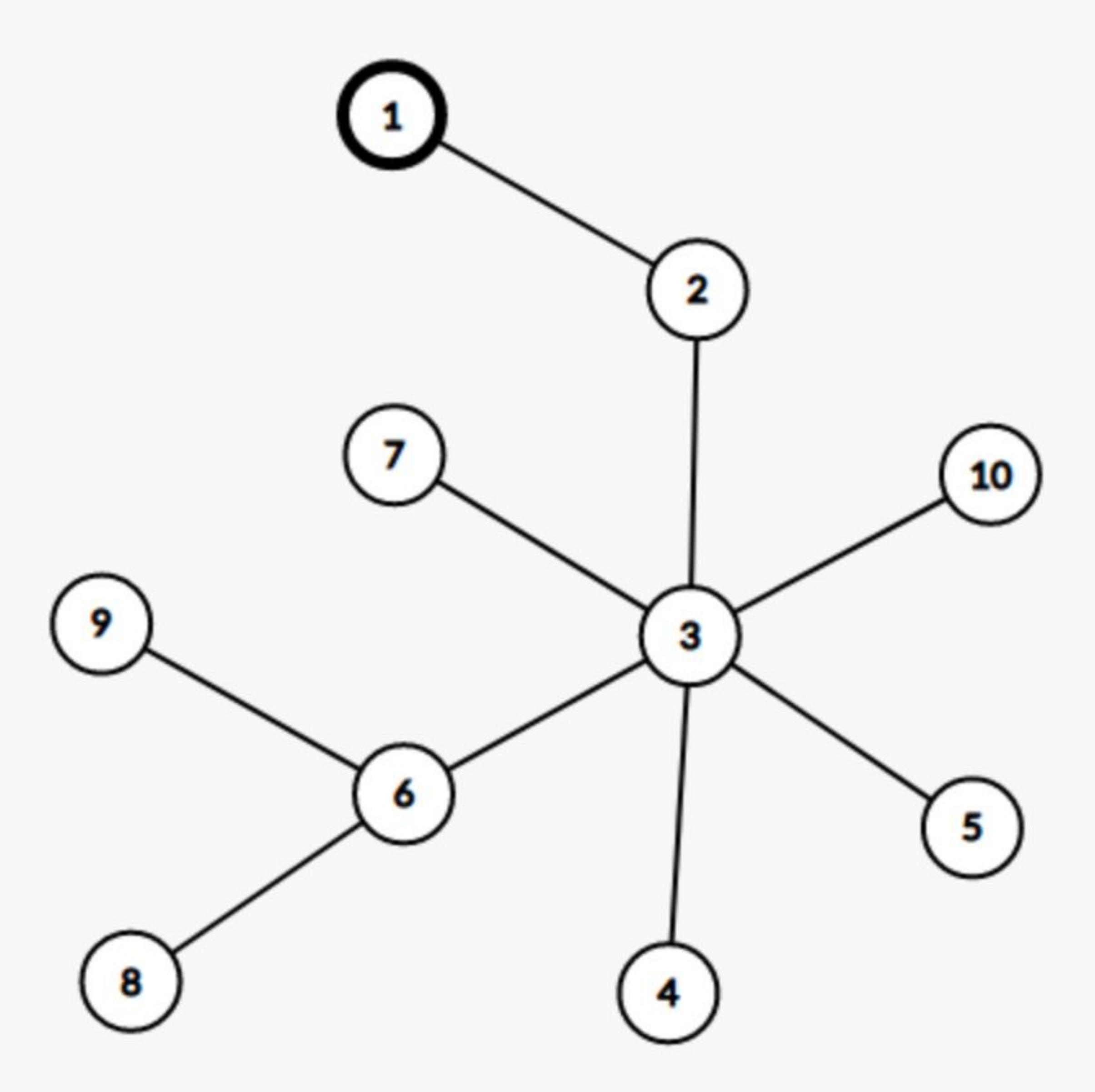
只要求出两点的LCA，就可以O(1) 计算两点之间的边权和。

2602树的直径

题目描述:

一棵树的直径就是这棵树上存在的最长路径。现在有一棵n个节点的树，现在想知道这棵树的直径包含的边的个数是多少?

如图所示的数据，这棵树的直径为(1-2-3-6-9)这条路径，包含的边的个数为4，所以答案是4。



2602树的直径

输入描述：

第1行：一个整数n，表示树上的节点个数。(1<=n<=100000)

第2-n行：每行有两个整数u,v,表示u与v之间有一条路径。(1<=u,v<=n)

输出描述：

输出一个整数，表示这棵树直径所包含的边的个数。

样例输入:
10
1 2
2 3
3 4
3 5
3 6
3 7
3 10
6 8
6 9
样例输出:
4

2602树的直径-解题思路1

我们不妨设1号点为根节点，那么这就可以看做一棵有根树。

设D[x]表示从节点x出发，往以x为根的子树走，能够到达的最远距离。设x的子节点分别为$y_1,y_2,y_3,\ldots,y_t$，

则可以得到状态转移方程：

$D[x]=Max(D[y_i])+1$

接下来，我们考虑对于每个节点x求出 **经过x的最长链的长度F[x]** ,整棵树的直径就是。

$Max(F[x])(1<=x<=n)$

2602树的直径-解题思路1

现在我们考虑如何求F[x] 。对于任意两个x的子节点y_i,y_j \ (i !=j) ，经过节点x的最长链的长度可以通过四个部分来构成：

1、 D[y_i]

2、D[y_j]

3、从x到y_i的距离（这个距离=1）

4、从x到y_j的距离（这个距离=1）

F[x]=Max(D[y_i]+D[y_j]+2)

所以我们只需要通过树形DP计算出D数组，之后对所有子节点的D取最大的两个向父节点转移即可。

2602树的直径-解题思路2

除了上面这种动态规划的方法，求树的直径还有一个简单而犀利的方法，使用2次DFS ，第一次随便一点R为根，找到离R 最远的一个点X，第二次DFS以X为根，找到离X最远的点Y ，X到Y的距离就是树的直径。该算法的正确性留给大家自己证明。