Problem A. Ring Road

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	1024 mebibytes

KOI City consists of N intersections and $N-1$ two-way roads. You can travel between two different intersections using only the given roads. In other words, the city's road network forms a tree structure. Roads are on a two-dimensional plane, and two roads do not intersect at locations other than the endpoints. Each road has an non-negative integer weight. This weight represents the time it takes to use the road.

KOI City was a small town until a few decades ago but began to expand rapidly as people arrived. In the midst of rapid expansion, the mayor had numbered the intersections between 1 and N for administrative convenience. The number system satisfies the following properties.

- Intersection 1 is the center of the city and is incident to at least 2 roads.
- The numbers assigned to intersections form one of the pre-orders of the tree rooted at intersection 1: for any subtree, the number of its root is the least number in that subtree.
- For each intersection, consider the lowest-numbered intersection among all adjacent (directly connected by road) intersections. When you list all adjacent intersections in a counterclockwise order starting from this intersection, the numbers go in increasing order.

With a large influx of people to KOI City, the traffic congestion problem has intensified. To solve this problem, the mayor connected the outermost cities with the *outer ring road*. Let $\{v_1, v_2, \ldots, v_k\}$ be the increasing sequence of numbers of all the intersections incident to exactly one road. For each $1 \le i \le k$, the mayor builds a two-way road between intersection v_i and intersection $v_{(i \bmod k)+1}$. The weight of each road is a nonnegative integer w_i. Due to the nature of the numbering system, you can observe that the outer ring road can be added in a two-dimensional plane in a way such that two roads do not intersect at any location except at the endpoint.

You are trying to build a navigation system for KOI city. The navigation system should answer Q queries of the form (u, v). For each query, the navigation system should return the shortest time it takes to move from intersection u to intersection v. The time to move through a path equals the sum of the weights of edges in the path.

Given a road network structure, write a program that efficiently answers Q queries.

Input

The first line contains the number of intersections N in the KOI City ($4 \le N \le 100\,000$).

Each of the next $N-1$ lines contains two space-separated integers p_i and c_i. They indicate that there is a two-way road with weight c_i connecting intersection p_i and intersection $i+1$ ($1 \le p_i \le i$, $0 \le c_i \le 10^{12}$).

Let k be the number of intersections incident to exactly one road in the original tree, and let $\{v_1, v_2, \ldots, v_k\}$ be the increasing sequence of their numbers. On the next line, k space-separated integers $w_1, w_2, \ldots, w_k$ are given. This indicates that the weight of the outer ring road connecting the intersection v_i and intersection $v_{(i \bmod k)+1}$ is w_i ($0 \le w_i \le 10^{12}$).

The next line contains the number of queries Q ($1 \le Q \le 250\,000$).

Each of the next Q lines contains two integers u and v denoting the intersections of interest ($1 \le u, v \le N$ and $u \ne v$).

Output

For each query, print a line with a single integer: the shortest time to move from u to v.

Examples

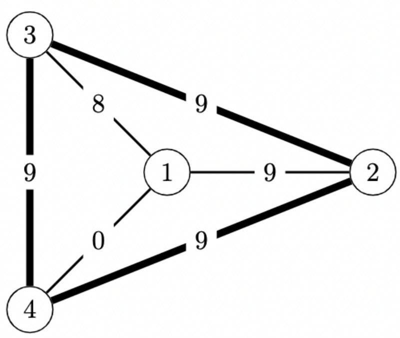
standard input	standard output
4	9
1 9	8
1 8	0
1 0	9
9 9 9	9
6	8
1 2	
1 3	
1 4	
2 3	
2 4	
3 4	

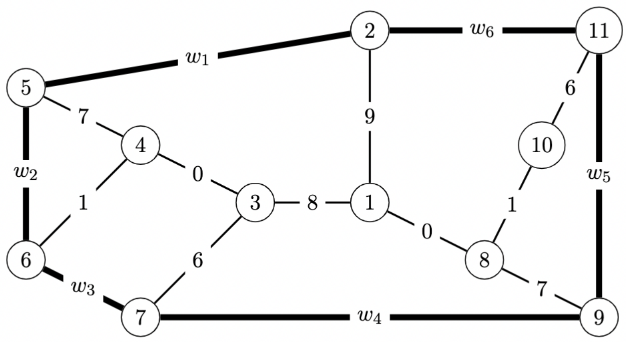
standard input	standard output
11	7
1 9	8
1 8	8
3 0	7
4 7	7
4 1	7
3 6	0
1 0	7
8 7	1
8 1	7
10 6	7
0 0 0 0 0 0	7
21	1
1 2	7
1 3	0
1 4	7
1 5	0
1 6	8
1 7	1
1 8	6
1 9	0
1 10	
1 11	
7 1	
8 2	
9 3	
10 4	
11 5	
1 6	
2 7	
3 8	
4 9	
5 10	
6 11	

standard input	*standard output*
11 1 9 1 8 3 0 4 7 4 1 3 6 1 0 8 7 8 1 10 6 1000000000000 1000000000000 1000000000000 1000000000000 1000000000000 1000000000000 21 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 1 11 7 1 8 2 9 3 10 4 11 5 1 6 2 7 3 8 4 9 5 10 6 11	9 8 8 15 9 14 0 7 1 7 14 9 15 9 22 9 23 8 15 16 16

Note

In the third sample, the line with $w_1, w_2, \ldots, w_k$ (in red) is split into several lines for readability.





The picture on the left corresponds to the first sample. The picture on the right corresponds to the second and third samples.