Problem G. Random Walking

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

In Berland, there are n cities and $n - 1$ distinct bidirectional roads connecting the cities. The cities are numbered from 1 to n, and there is a way along the roads between each pair of cities (so, Berland can be considered a tree).

When a tourist flies from other country to the city u and wants to visit the city v, he repeats the following procedure until he reaches the city v: the tourist chooses a *neighbor city* (that is, the city which is connected by a road with the city our tourist is located in) and moves to it. Traveling every road between adjacent cities takes exactly one hour. As Berland has no signposts, the tourist does not know where to go, so each time he selects a neighbor, the selection is random, and all neighbors are equally probable. Note that the tourist does not remember the roads he traveled before, and thus can choose the road he used to arrive in the city as well as every other road.

The king of Berland is concerned about the average time spent by a tourist, so he wants to know how long it takes (on average) to reach v_i starting in u_i for some pairs u_i and v_i. Answer his questions.

Input

The first line contains one integer n, the number of cities in Berland ($1 \le n \le 10^5$).

Next $n - 1$ lines contain descriptions of the roads as pairs of integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) meaning that there is a road between a_i and b_i. It is guaranteed that the given graph is a tree.

The following line contains the only integer q, the number of questions ($1 \le q \le 2 \cdot 10^5$).

Next q lines describe the king's questions: i-th of these lines contains two space-separated integers u_i and v_i ($1 \le u_i, v_i \le n$).

Output

For each question, print the average time (in hours) a tourist will spend to get from u_i to v_i. The answer is considered to be correct if its absolute or relative error does not exceed 10^{-9}.

Example

standard input	standard output
3 1 2 2 3 3 1 2 2 1 3 3	1.000000 3.000000 0.000000