New Year Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

The first qualifying round of **Innopolis Open** had finally ended. The results were finalized, the participants' questions were answered, and for the first time in quite a while the organizers could get some rest.

Meanwhile, New Year was approaching. Festive lights were appearing in Kazan, New Year trees were being decorated in the streets, and only the organizers' room still had no decorations at all.

— We urgently need a New Year tree! — Tatyana announced.

— But an unusual one, — Renata added. — This is Innopolis, after all!

After a short discussion, the managers decided to build their own **two-dimensional New Year tree**. Mefodiy took charge of the wire frame, Diana and Vlada helped assemble the structure, and the others worked on ornaments, lights, and decorations.

The result was an unusual tree-shaped installation: vertices became places to attach ornaments, and edges became wire branches.

But then the organizers realized that the holiday budget was almost exhausted. To display the installation properly, they needed a rectangular frame and wanted to use as little material for it as possible.

Things were complicated by Mefodiy's adjustable joints: any branch, together with all its offshoots, could now be rotated or reflected!

The managers enjoyed experimenting with the tree, but after each change they had to recalculate the size of the frame.

Fortunately, just then Ilnur and Seva stopped by, having finally caught up on their sleep after the first qualifying round.

Seeing the unusual construction and the piles of handwritten calculations, they immediately realized that this was not merely a holiday problem, but a perfect task for the second round of Innopolis Open.

And now it is your turn to solve it!

Formal Statement

You are given a tree with n vertices, rooted at vertex 1. Each vertex v is initially located at point (x_v, y_v) in the plane.

Process q queries of four types:

1. `1 v k` — rotate all vertices in the subtree rooted at v counterclockwise by $45k^\circ$ around the current position of vertex v, where $1 \le k \le 7$.
2. `2 v` — reflect all vertices in the subtree rooted at v across the horizontal line passing through the current position of vertex v.
3. `3 v` — reflect all vertices in the subtree rooted at v across the vertical line passing through the current position of vertex v.
4. `4 v` — find the perimeter of the smallest axis-aligned rectangle containing all vertices in the subtree rooted at v at their current positions.

All transformations are performed sequentially. The tree structure and the lengths of its edges do not change. After transformations, edges may intersect and vertices may coincide. The rectangle may be degenerate: its perimeter is always twice the sum of its width and height.

Input

The first line contains two integers n and q ($1 \le n, q \le 2 \cdot 10^5$) — the number of vertices and queries.

Each of the next n lines contains two integers x_v and y_v ($-10^6 \le x_v, y_v \le 10^6$) — the initial coordinates of the vertices, in order from 1 to n.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$) — the endpoints of an edge. It is guaranteed that these edges form a tree.

Each of the next q lines contains one of the following queries:

- `1 v k`, where $1 \le v \le n$ and $1 \le k \le 7$;
- `2 v`, where $1 \le v \le n$;
- `3 v`, where $1 \le v \le n$;
- `4 v`, where $1 \le v \le n$.

Output

For each query of type 4, output a single real number on a separate line: the minimum perimeter of the rectangle enclosing the specified subtree. Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Scoring

Points for a subtask are awarded only if the solution successfully passes all tests of this subtask and all subtasks listed in the column "Required subtasks".

In each subtask, it is necessary to answer correctly all queries of type 4.

Subtask	**Points**	**Additional constraints**		**Required subtasks**
		n, q	**Other conditions**	
0	0	Examples from the statement		—
1	5	$n, q \le 8 \cdot 10^4$	All queries are of type 4.	—
2	5	$n \le 30, q \le 8 \cdot 10^4$	—	—
3	6	$n, q \le 2000$	—	—
4	7	$n, q \le 8 \cdot 10^4$	The number of queries of types 1, 2, and 3 does not exceed 8.	1
5	8		For all queries of types 1, 2, and 3, $v = 1$.	1
6	9		The sum of subtree sizes over all queries of types 1, 2, and 3 does not exceed $2 \cdot 10^5$.	1
7	6		Only queries of types 2, 3, and 4 occur.	1
8	6		Only queries of types 1 and 4 occur; in queries of type 1, $k \in \{2, 4, 6\}$.	1
9	9		All query types are allowed; in queries of type 1, $k \in \{2, 4, 6\}$.	1, 7, 8
10	11		Only queries of types 1 and 4 occur.	1, 8
11	8		No additional constraints.	1–10
12	20	$n, q \le 2 \cdot 10^5$	No additional constraints.	1–11

Example

standard input	standard output
4 11 0 0 2 0 2 2 0 2 1 2 1 4 2 3 4 1 2 2 4 2 1 2 2 4 1 3 1 4 2 1 1 1 4 1 4 3 4 2	8.0000000000 4.0000000000 12.0000000000 4.0000000000 14.1421356237 0.0000000000 5.6568542495

Note

Initially, all vertices are located in the square with opposite vertices $(0, 0)$ and $(2, 2)$, so the first answer is $2(2 + 2) = 8$.

After reflecting the subtree of vertex 2 across the horizontal line, vertex 3 moves from $(2, 2)$ to $(2, -2)$. The perimeter of the frame of subtree 2 is 4.

After rotating the subtree of vertex 2 by $90°$, vertex 3 ends up at point $(4, 0)$, and the perimeter of the frame of the whole tree is 12. A subsequent reflection of the whole tree across the vertical line $x = 0$ does not change the perimeter of subtree 2: it is still 4.

After rotating the whole tree by $45°$, the perimeter of the frame for all vertices is $10\sqrt{2} \approx 14.1421356237$. For the subtree of leaf 3, the perimeter is 0, and for the subtree of vertex 2, it is $4\sqrt{2} \approx 5.6568542495$.



New Year's installation of Innopolis Open.