



Task 4: Grapevine

Syrup the Turtle waters the huge Grapevine which snakes around town. The layout of the Grapevine can be best described as having N leafy joints, which Syrup has numbered 1 to N , connected by $N - 1$ branches also numbered 1 to $N - 1$. Each branch i directly connects two joints A_i and B_i , and has a length of W_i . No two branches directly connect the same pair of joints, and as part of the single Grapevine, every joint is connected to every other either directly or indirectly by branches.

With his sturdy hose and a little handiwork, Syrup is able to sway the growth of the Grapevine as he deems fit. In particular, he can **soak** a joint of the vine - causing it to immediately sprout a single giant grape if it was empty, or instead dislodging the grape there if one was present.

He can also **anneal** a branch with water, extending or shortening it to a new length w_i by spraying at the right pressure and angle. To make sure things are on track, Syrup will periodically stand atop an elevated joint and **seek** for the closest grape. The distance from such a joint to a grape is defined by the shortest path starting from said joint, traversing a number of branches (or none at all), and ending at the grape's own joint.

Right now, the Grapevine has no grapes attached after the last passing storm. Syrup has his watering agenda of Q actions planned out for the week and is about to begin spraying; but first, he needs to know what distances to expect when he **seeks** for grapes along the way. Given Syrup's watering plans, your task is to find for each planned **seek** the distance from the specified joint to the nearest grape.

Input

Your program must read from standard input.

The first line contains two integers, N and Q .

$N - 1$ lines will follow. The i^{th} line contains three integers, A_i , B_i , and W_i , describing a single branch.

Q lines will follow, each representing an action by Syrup.

- If the first integer of the line is 1, it represents a **seek** and 1 integer q_i will follow. This means that you are to determine the distance between joint q_i and the nearest joint with a grape, and output this distance. If there are no grapes on the Grapevine, output -1 instead.
- If the first integer of the line is 2, it represents a **soak** and 1 integer u_i will follow. This



means that joint u_i is soaked and grows a grape or has its grape dislodged.

- If the first integer of the line is 3, it represents an **anneal** and 3 integers a_i , b_i , and w_i will follow. This means that the length of the branch between joints a_i and b_i has had its length changed to w_i . It is guaranteed that a branch between joints a_i and b_i exists.

Output

Your program must print to standard output.

For each **seek** action, output one line with a single integer, the distance to the closest grape, or -1 if no grapes are present.

Subtasks

The maximum execution time on each instance is 3.0s. For all testcases, the input will satisfy the following bounds:

- $2 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq A_i \neq B_i \leq N$
- $0 \leq W_i \leq 10^9$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Points	Additional Constraints
1	6	$N, Q \leq 2000$
2	14	For all seek actions, $q_i = 1$
3	15	The vine forms a complete binary tree, $A_i = \lfloor \frac{i+1}{2} \rfloor$, $B_i = i + 1$
4	15	There is at most 1 grape on the vine at any point in time.
5	18	All soak actions will occur before any seek or anneal actions. For all anneal actions, $w_i = 0$
6	32	-



Sample Testcase 1

This testcase is valid for subtasks 1, 2, 5, and 6.

Input	Output
7 11	11
1 2 2	8
2 3 4	8
5 6 1	2
5 3 6	
3 7 6	
2 4 9	
2 6	
2 4	
2 7	
1 1	
3 2 3 0	
1 1	
3 6 5 0	
1 1	
3 3 5 0	
3 2 4 0	
1 1	

Sample Testcase 1 Explanation

In the first **seek**, the closest grape is at joint 4 over the path $1 \rightarrow 2 \rightarrow 4$, for a distance of $2 + 9 = 11$. The second **seek** has the closest grape at joint 7, while the third **seek** has both the grapes at joints 6 and 7 closest. The fourth and final **seek** has its closest grape at joint 4.

Sample Testcase 2

This testcase is valid for subtasks 1, 3, and 6.



Input	Output
6 11	7
1 2 3	2
1 3 4	-1
2 4 1	4
2 5 4	
3 6 6	
2 3	
1 2	
2 4	
3 1 3 2	
1 1	
2 3	
3 2 1 2	
2 4	
1 3	
2 2	
1 3	

Sample Testcase 3

This testcase is valid for subtasks 1, 4, and 6.

Input	Output
7 8	3
1 2 2	11
2 3 3	0
3 6 2	8
4 6 1	
5 6 4	
6 7 3	
2 3	
1 4	
2 3	
2 5	
1 1	
3 6 7 4	
1 5	
1 7	