Module 3: Routing with Traffic Lights

Problem Description

Time Limit: 1 second
Memory Limit: 256 MB

Another issue that drivers face along their route is the red lights at intersections.

TapNap wants to address this issue in its router and determine the path such that the travel time from intersection 1 to n is minimized.

Note that, similar to the previous section, every road consumes time equal to its length (edges take time to traverse).

Assume that the duration of the green light and red light for each intersection is known. Also assume that the lights have two states, green and red. They will be green for a duration of t_{on} and red for a duration of t_{off}. At time zero, all lights are green and will remain green for t_{on}. For example, if we reach an intersection at any time from 0 to t_{on} (exclusive of the exact moment the light turns red), we can pass through it without waiting.

You are expected to produce a routing program that finds the shortest time to reach intersection n from intersection 1, considering only the traffic lights at the intersections.

Clarification: A cycle typically lasts $t_{on} + t_{off}$. The light is green in the interval $[0, t_{on})$ modulo the cycle duration. If you arrive during the red interval, you must wait until the start of the next green interval.

Input Format

- The first line contains two integers n and m, representing the number of intersections and the number of two-way streets, respectively.
- In each of the next m lines, three integers u, v, and w are given, representing the two ends of the street and the length of the street (travel time).
- The last n lines of the input describe the traffic light conditions. Each line corresponds to an intersection (from 1 to n). Each line contains two numbers t_{on} and t_{off}, representing the duration the light is green and the duration it is red, respectively.

Constraints

- $1 \leq n, m \leq 2 \times 10^5$
- $1 \leq t_{on}, t_{off} \leq 10^6$
- Edge weights $w \leq 10^9$

Output Format

Print a single number in the only line of output: the minimum time required to reach intersection n from intersection 1.