Problem Apr 20: Easter

Time limit: 2 seconds

Image generated by OpenAI's DALL-E.

Keeping the great Easter egg factory running is hard work. After all, all Easter eggs need to be ready in time for Easter, so that every child has something to look forward to. As a logistics bunny, your job is transporting things between the different parts of the factory so that everything runs smoothly.

Every morning, you get assigned a fixed route consisting of n stops that you have to visit in the given order. For each stop, you also get a time window $[s, e]$ during which you have to visit the stop to make deliveries and pick up new items, as otherwise things might not be ready for you or the production line might back up. You also know the time d that it takes you to run there from the previous stop (or from your home in case of the first stop, where you start at time 0).

As there is a constant stream of logistics bunnies visiting every part of the factory, you may only visit each stop once in the time window. Luckily, you are fast enough so that the time you spent loading and unloading at each stop is negligible. The route optimization bunnies that assign you your route ensure that you can always make it to every stop in time. You might even have to take some pauses if you arrive at some stops before the start of their respective time windows.

However, this year there is a new complication: as Easter falls quite late this year, the sun is already shining down with quite some force. This is no problem for you, however, the chocolate eggs you are transporting might melt if you are not careful. You made a list of m pairs of stops $a \to b$ that you are transporting chocolate parts between, each with a time l that you think the parts can be transported by you without melting. Is there a way to strategically make pauses during your route to ensure that all chocolate parts arrive intact, while still reaching all your stops in time?

Input

The input consists of:

- One line containing two integers n and m ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$), the number of stops and the number of chocolate transports.
- n lines, each containing three integers s, e, and d ($0 \leq s \leq e \leq 10^9$, $1 \leq d \leq 10^9$), the earliest and latest time you can visit the stop (both inclusive), and the time it takes you to run to the stop from the previous stop/your home.
- m lines, each containing three integers a, b, and l ($1 \leq a < b \leq n$, $1 \leq l \leq 10^9$), indicating that you must get from stop a to stop b in at most l time.

It is guaranteed that you would be able to reach every stop in time if the chocolate parts melting was not a problem.

Output

If there is no way to run your route without any chocolate melting or being late for a stop, output "`impossible`". Otherwise, output "`possible`" followed by n integers: the time at which you visit each stop, in the order that you visit them.

Sample Input 1

```
2 1
0 2 1
4 6 2
1 2 2
```

Sample Output 1

```
possible
2 4
```

Sample Input 2

```
3 1
0 2 1
4 9 1
5 6 2
1 3 3
```

Sample Output 2

```
impossible
```

Sample Input 3

```
4 2
1 2 1
0 5 2
5 8 1
8 10 2
1 3 3
2 4 3
```

Sample Output 3

```
impossible
```