Temporal Graph Shortest Path With Time-Dependent Edges

Time Limit: 1 second
Memory Limit: 256 MB

Problem Description

We are given a directed temporal weighted graph with N nodes and M directed edges. Each edge e is represented as a tuple (u, v, w, L, R) meaning there is a directed edge from node u to node v that takes exactly w time units to traverse but is only usable when the traversal starts at a time t with $L \leq t \leq R$. If you arrive at node u at time t and $t < L$, you may wait at node u until time L and then traverse the edge; if $t > R$, the edge cannot be used.

Starting at time 0 from source node S, compute the earliest possible arrival time at target node T. If T is unreachable under the temporal constraints, output `-1` or state "No path".

Input format

```
N M
S T
M lines follow:
u v w L R
```

All values are integers. Nodes are 1-indexed.

Output format

Single line:

```
<Earliest arrival time>
```

or

```
No path
```

Constraints

- $1 \leq N \leq 100{,}000$.
- $0 \leq M \leq 200{,}000$.
- $0 \leq L \leq R \leq 10^9$.
- $1 \leq w \leq 10^6$.

Sample inputs and outputs

Sample 1: Input:

```
5 6
1 5
1 2 3 0 10
2 3 4 5 20
3 5 2 0 100
1 4 5 3 6
4 3 3 6 8
2 5 20 5 7
```

Output:

```
11
```

Explanation: One feasible arrival at time 11 is via 1→2 arrive at 3, wait until 5, 2→3 arrives at 9, 3→5 arrives at 11.

Sample 2: Input:

```
3 2
1 3
1 2 5 10 20
2 3 5 0 4
```

Output:

```
No path
```

Explanation: Edge (1→2) only opens at 10, but (2→3) closes at 4, so cannot proceed.