There is a funny car racing in a city with n junctions and m directed roads.

The funny part is: each road is open and closed periodically. Each road is associate with two integers (a, b), that means the road will be open for a seconds, then closed for b seconds, then open for a seconds... All these start from the beginning of the race. You must enter a road when it's open, and leave it before it's closed again.

Your goal is to drive from junction s and arrive at junction t as early as possible. Note that you can wait at a junction even if all its adjacent roads are closed.

Input

There will be at most 30 test cases. The first line of each case contains four integers n, m, s, t ($1 \leq n \leq 300$, $1 \leq m \leq 50,000$, $1 \leq s, t \leq n$). Each of the next m lines contains five integers u, v, a, b, t ($1 \leq u, v \leq n$, $1 \leq a, b, t \leq 10^5$), that means there is a road starting from junction u ending with junction v. It's open for a seconds, then closed for b seconds (and so on). The time needed to pass this road, by your car, is t. No road connects the same junction, but a pair of junctions could be connected by more than one road.

Output

For each test case, print the shortest time, in seconds. It's always possible to arrive at t from s.

Sample Input

```
3 2 1 3
1 2 5 6 3
2 3 7 7 6
3 2 1 3
1 2 5 6 3
2 3 9 5 6
```

Sample Output

```
Case 1: 20
Case 2: 9
```