Bus Trip (Estonia)

TASK

There are N towns, and M one-way direct bus routes (no intermediate stops) between them. The towns are numbered from 1 to N. A traveler who is located in the town 1 at time 0 needs to arrive in the town P. He will be picked from the bus station at town P exactly at time T. If he arrives earlier he will have to wait.

For each bus route i, we know the source and destination towns s_i and t_i, of course. We also know the departure and arrival times, but only approximately: we know that the bus departs from s_i within the range $[a_i, b_i]$ and arrives at t_i within the range $[c_i, d_i]$ (endpoints included in both cases).

The traveler does not like waiting, and therefore is looking for a travel plan which minimizes the maximal possible waiting time while still guaranteeing that he'll not miss connecting buses (that is, every time he changes buses, the latest possible arrival of the incoming bus must not be later than the earliest possible departure time of the outgoing bus).

When counting waiting time we have to assume the earliest possible arrival time and the latest possible departure time.

Write a program to help the traveler to find a suitable plan.

INPUT

The input file name is `TRIP.IN`. The first line contains the integer numbers N ($1 \le N \le 50{,}000$), M ($1 \le M \le 100{,}000$), P ($1 \le P \le N$), and T ($0 \le T \le 1{,}000{,}000{,}000$).

The following M lines describe the bus routes. Each line contains the integer numbers s_i, t_i, a_i, b_i, c_i, d_i, where s_i and t_i are the source and destination towns of the bus route i, and a_i, b_i, c_i, d_i describe the departure and arrival times as explained above ($1 \le s_i \le N$, $1 \le t_i \le N$, $0 \le a_i \le b_i < c_i \le d_i \le 1{,}000{,}000{,}000$).

OUTPUT

The only line of the output file `TRIP.OUT` should contain the maximal possible total waiting time for the most suitable possible travel plan. If it is not possible to guarantee arrival in town P by time T, the line should contain –1.

EXAMPLES

INPUT

```
3 6 2 100
1 3 10 20 30 40
3 2 32 35 95 95
1 1 1 1 7 8
1 3 8 8 9 9
2 2 98 98 99 99
1 2 0 0 99 101
```

OUTPUT

```
32
```

The most pessimistic case for the optimal travel plan for the above example is as follows:

Time	**Action**
0…1	Wait in town 1
1…7	Take the bus line 3 from town 1 to town 1
7…8	Wait in town 1
8…9	Take the bus line 4 from town 1 to town 3
9…35	Wait in town 3
35…95	Take the bus line 2 from town 3 to town 2
95…98	Wait in town 2
98…99	Take the bus line 5 from town 2 to town 2
99…100	Wait in town 2

Total waiting time: 1+1+26+3+1=32

INPUT

```
3 2 2 100
1 3 0 0 49 51
3 2 50 51 100 100
```

OUTPUT

```
-1
```