Infinathlon

time limit per test: 2 seconds (10 seconds for Python)
memory limit per test: 1 gigabyte
input: standard input
output: standard output

points: 100 · problem code: polyathlon · author: glipR

The MAPS Committee is looking to spice things up with the Pentathlon and introduce some more sports.

The "Infinathlon" is a new sporting event that has competitors running between different arenas, completing different sports, with the aim of reaching the final arena.

How this works is that each competitor starts off with 0 seconds on their timer, and whenever they are running between arenas, this timer decrements. If the timer ever goes below 0 seconds, then they are disqualified.

Within the arenas, competitors can do challenges from other sporting events to earn time on their clock, allowing them to travel between arenas. Depending on the strength of their performance, for every second they add to their clock, they'll accrue a certain amount of penalty points. Finally, the clock also maxes out at a certain time-limit, stopping competitors from filling out their clock at a single event, and encouraging competitors to use multiple arenas on their journey.

The winner of the competition is not the fastest competitor, but the one that accrues the *least penalty points*.

You're trying to help out your friend Eric. Eric's been given the map layout for the Infinathlon ahead of time, and Eric knows his strengths and weaknesses, in particular, he knows:

- How long it will take him to run between arenas, in exact seconds, and
- For every arena, how many penalty points he will get per second added to his timer there.

Eric would like to know the minimum number of penalty points to get him from arena s (with 0s on the clock) to arena t.

Input

Input will begin with 5 integers:

1. The number of arenas n ($1 \leq n \leq 200$)
2. The number of pathways between arenas m ($0 \leq m \leq 39800$)
3. The maximum number of seconds you can have on your clock ($0 \leq c \leq 10^9$)
4. The starting arena s ($1 \leq s \leq n$)
5. The ending arena t ($1 \leq t \leq n$)

The next line will contain n integers p_i ($1 \leq p_i \leq 10^6$), representing the amount of penalty points you know you'll get per second earned in each arena, from 1 to n.

m lines will follow, detailing the paths between arenas. Note that these paths are **directed**.

Each line will contain 3 integers x, y and z, representing that travelling from arena x to arena y will take exactly z seconds ($1 \leq x, y \leq n$, $1 \leq z \leq 2 \times 10^9$).

There are no self loops ($x \neq y$), and no directed edge appears in the input twice.

Output

Output a single integer, representing the minimal penalty required to get from arena s (Starting with 0s) to arena t.

If it is impossible to reach arena t, then instead output the string `Unreachable`

Example 1

Input

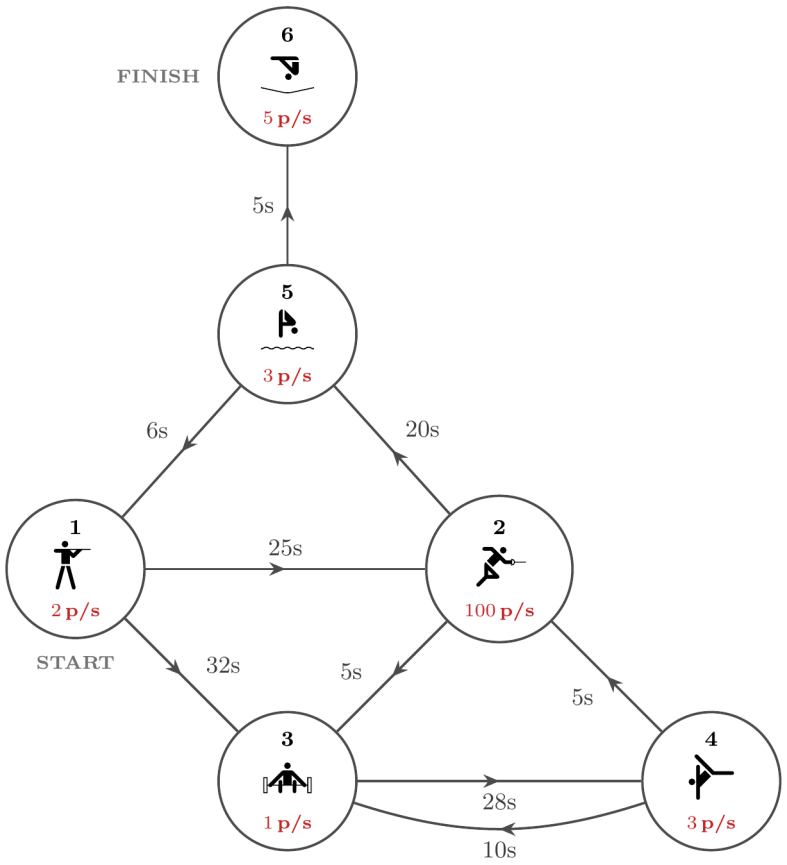
```
6 9 30 1 6
2 100 1 3 3 5
1 2 25
1 3 32
2 3 5
3 4 28
4 3 10
4 2 5
2 5 20
5 1 6
5 6 5
```

Output

```
174
```

Explanation

The sample input represents the following field:

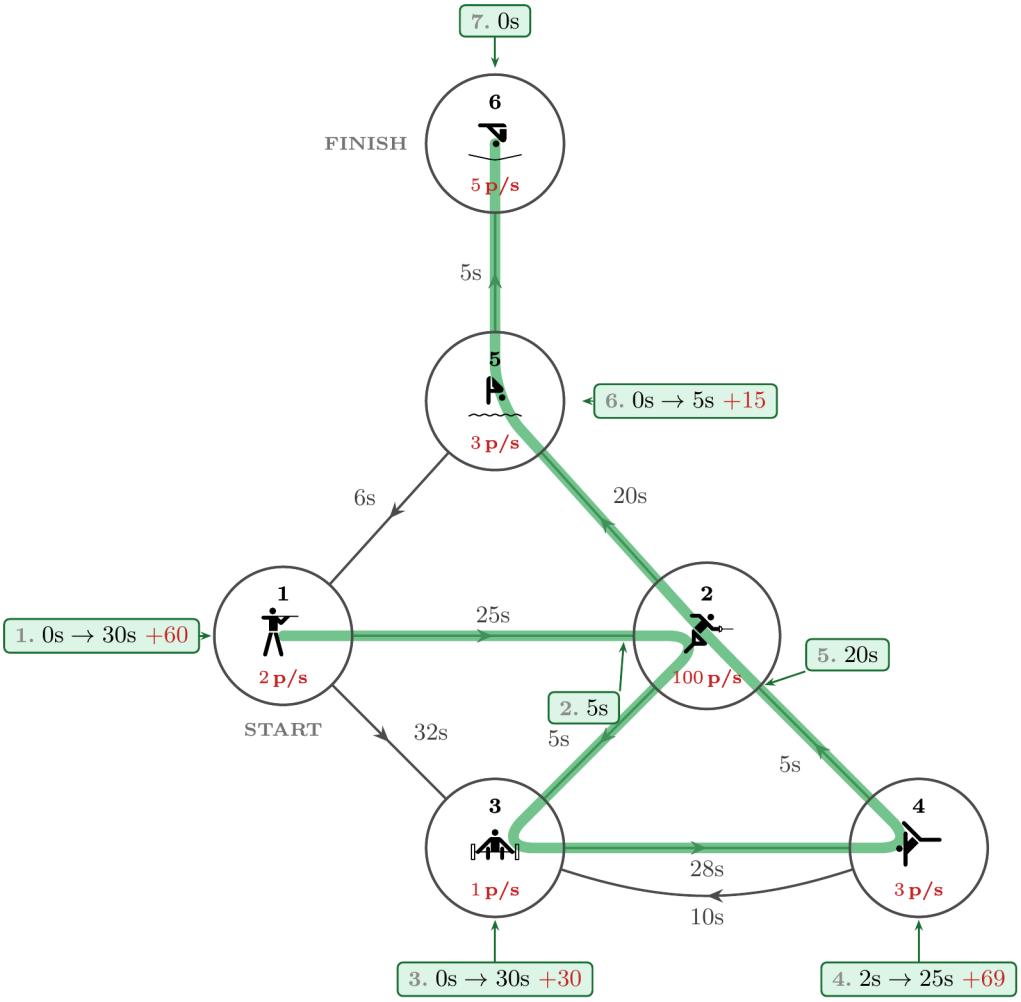


6 arenas, 9 pathways. Clock capacity $c = 30$s.
Start at arena 1, finish at arena 6.

Which can be solved in 174 penalty by:

1. Immediately maxing out 30s on the clock at Node 1: Shooting (+60 penalty)
2. Running from Shooting(1) to Fencing(2), and Fencing(2) to Weightlifting(3)
3. Maxing out again at Weightlifting (+30 penalty)
4. Running to Breakdancing(4)

5. Earning 23s at Breakdancing(4) (+69 penalty)
6. Running to Fencing(2) then Diving(5)
7. Earning 5s at Diving(5) (+15 penalty)
8. Running to Trampoline Gymnastics(6), the goal.



Total penalty $= 60 + 30 + 69 + 15 = \mathbf{174}$.