Find Rooms For Events

Learning new exciting things never ends in Reply! There are always lot of events – from short hands-on seminars (Labcamps) and full-day deep dives (Bootcamps), to conventions (Netcamps) and even weekend-long code camps (Hackathons).
In fact, there are so many, that finding rooms for them is a challenge!
Do you mind giving us a hand?

Problem Statement
Given the number of events ‘E’, and the number of rooms available for hosting such events ‘R’, your task is to create the best way of scheduling events to maximise overall learning and efficient room use.
Since the number of events can exceed the number of rooms, it’s fine to leave some events unscheduled.
Each event is characterised by its topic, start time, end time, and number of participants. Each room has a name and a capacity.
Please create a schedule of events to run in each room, so that the total number of participants in all events and the total time the rooms are actually occupied are as high as possible.
Each room can only host events whose number of participants is less than or equal to its capacity.
Each event can only take place in one room and each room can only host one event at the time: two events cannot take place at the same time, in the same room.

Input
Input data will be provided in a text file. On the first line you will find two integer E and R, representing the number of events and rooms you'll find in the file.
E lines follows, each describing a single event. Those are composed of 4 values separated by space:

1. A string representing the event name.
2. An integer representing the UNIX timestamp of the starting time.
3. An integer representing the UNIX timestamp of the ending time.
4. An integer p representing how many participants registered for the event.

R lines describe the rooms. Each line consists of two space-separated values:

1. A string representing the room name.
2. An integer ‘c’ representing how many people the room can host.

Please note that all rooms open when the first event starts, and they all close when the last event ends.

Input example:

```
8 3
cereal-minds 1494063000 1494070200 30
code-for-kids 1494063000 1494095400 100
angular-labcamp 1494072000 1494083700 25
aws-webinar 1494086400 1494093600 50
secuity-bootcamp 1494070200 1494081000 20
springboot-labcamp 1494082800 1494090000 15
aperitime 1494091020 1494095400 20
student-tech-clash 1494061200 1494068400 5
solar 80
mini-conference 30
solar-garden 100
```

Constraints

$1 \le E \le 1000000$

$1 \le R \le 100000$

$0 \le p \le 100$ (event participants)

$1 \le c \le 100$ (room capacity)

Output

R lines, each one consisting of the name of the room, a colon, and the space-separated list of events hosted in that room. You must list events in the same order they will be held. So if one event has to happen before another one, it must precede the other one on the list. It’s okay for an event to start immediately after the previous one ends (previous.endTime == next.startTime).

The relative order of rooms is not important – any order is acceptable.

Output example:

```
solar:student-tech-clash secuity-bootcamp aws-webinar
mini-conference:cereal-minds angular-labcamp aperitime
solar-garden:code-for-kids
```

Scoring

Your solution will be checked against the rules described above before being graded. Only valid solutions will receive a score.

Each scheduled event will get a score calculated as the ratio between that event's participants over its assigned room capacity, multiplied by the event duration in seconds.

A score will be assigned for each room in the input file, calculated as the sum of the scores of their assigned events, minus an amount proportional to the room unallocated time and calculated as: *(room capacity / the capacity of the biggest room) x (total room opening time — total duration of scheduled events)*.

The final score assigned to your submitted file will be given by the sum of all room's scores.

The best scores you will achieve on each input file will be summed up in order to assign your final challenge score that will determine the leaderboard.

Complete Example

Input example:

```
8 3
cereal-minds 1494063000 1494070200 30
code-for-kids 1494063000 1494095400 100
angular-labcamp 1494072000 1494083700 25
aws-webinar 1494086400 1494093600 50
secuity-bootcamp 1494070200 1494081000 20
springboot-labcamp 1494082800 1494090000 15
aperitime 1494091020 1494095400 20
student-tech-clash 1494061200 1494068400 5
solar 80
mini-conference 30
solar-garden 100
```

Output example:

```
solar:student-tech-clash secuity-bootcamp aws-webinar
mini-conference:cereal-minds angular-labcamp aperitime
solar-garden:code-for-kids
```

Score
Below you can find a detail of the score calculation.

The capacity of the biggest room is calculated as: max (solar, mini-conference, solar-garden) = max (80, 30, 100) = 100

solar events scoring: (5/80 * 7200) + (20/80 * 10800) + (50/80 * 7200) = 450 + 2700 + 4500 = 7650
solar occupation scoring: 80/100 * (34200 - 7200 - 10800 - 7200) = 7200
solar score: 7650 - 7200 = **450**

mini-conference events scoring: (30/30 * 7200) + (25/30 * 11700) + (20/30 * 4380) = 7200 + 9750 + 2920 = 19870
mini-conference occupation scoring: 30/100 * (34200 - 7200 - 11700 - 4380) = 3276
mini-conference score: 19870 - 3276 = **16594**

solar garden events scoring: (100/100 * 32400) = 32400
solar garden occupation scoring: 100/100 * (34200 - 32400) = 1800
solar garden score: 32400 - 1800 = **30600**

This output would score 47644 points (**450** + **16594** + **30600** = 47644)