Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Prof. Dr. Friedrich, Dr. Lenzner, Boockmeyer, Neumann, Stangl
Sommersemester 2017

2. Contest – (Adv.) Competitive Programming

Abgabe 20.06.17 15:30 - 18:30 Uhr, über das Judge-Interface

research-servers: (100 Points - 1 second timelimit)

The HPI decided to invest into some new powerful servers specifically for research purposes. They will be used by researchers, who can request server time, and, if not used otherwise, for a long-running research calculation.

For this calculation, multiple servers are sorted into a list, and only communicate with the server directly before and after them in the list. Servers can join in on the calculation at any time, but have to know every server they will communicate with beforehand.

Setting this up is your job. Each week you get a timetable for every server, containing the timeslots during which it will take part in the calculation. Now you have to figure out which pairs of servers will communicate with each other.

Servers have continuous ids ≥ 0, and are sorted into the server list based on them. Therefore, if server 1 and 5 are calculating and server 2 joins in, the list is now $(1, 2, 5)$, with 1 and 2, as well as 2 and 5 communicating, but not 1 and 5.

Timeslots are given as a start and endtime. Both are points in time in seconds (if $\mathit{endtime} - \mathit{starttime} = x$, the slot is x seconds long). For servers to be able to communicate they need to run in parallel for at least one second.

Input The input begins with a line containing w $(0 \leq w \leq 10)$, the number of weeks to process. Each week starts with a line containing s $(0 \leq s \leq 10000)$, the number of servers to configure, and t $(0 \leq t \leq 60000)$, the number of timeslots. The following t lines each contain i $(0 \leq i < s)$, the id of the server, as well as t_s and t_e $(0 \leq t_s, t_e \leq 604800; t_s < t_e)$, the start and end time of the timeslots. No two timeslots for the same server will overlap.

Output For each week, first output the number of pairs that follow. Then output a list of pairs a, b (one line per pair), indicating that server a and b need to communicate during that week. Only output the pairs with $a < b$. Output the pairs sorted by a, and, if a is equal, by b.

Points There are three groups of test sets:

- *easy:* The first group of operations worth 20 points, you can assume that $t_e = 604800$ for each timeslot.
- *medium:* For the second group worth 30 Points, you can assume that $t < 17000$.
- *hard:* For the third group worth 50 Points, there are also no additional assumptions.

Sample Input

```
1
4 4
0 0 20
1 5 15
2 12 14
3 17 19
```

Sample Output

```
3
0 1
0 3
1 2
```