Dave is a die-hard fan of a show called "HackerRank", in which a young programmer uses her problem-solving abilities to solve crimes. He splurged on a Digital Video Recorder (DVR) so that he can record HackerRank episodes and watch them later. Luckily, Dave managed to get his hands on schedules for all the episodes in each upcoming season.

Each season has n episodes numbered from 1 to n. Each episode airs twice; the first time it's called "live", and the second time it's called "repeat". So, for each episode, we have 4 integers, $(s_{live}$ and $e_{live})$ for the live airing and $(s_{repeat}$ and $e_{repeat})$ for the repeat airing, where s is episode's start time and and e is its end time. All times are given as integers representing the number of minutes passed since the start of the season.

Episodes broadcast on multiple channels, so some of the air times overlap and the episodes may not broadcast sequentially. It's possible that both the live and repeat broadcasts of some episode i are held before episode j, even though $i > j$. In addition, live and repeat broadcasts of the same episode may differ in length due to the number of advertisements during the broadcast.

Dave only has one TV with a DVR attached to it, and the DVR is capable of recording one episode at a time. For each episode in a season, Dave first decides whether or not he will record it. If he decides to record it, he will either record it during (s_{live}, e_{live}) or (s_{repeat}, e_{repeat}). Dave will only ever record one of the two airings of an episode, and he always records *full* episodes. This means that once he starts recording an episode, he will always record it until the end (i.e., he never records partial episodes).

Dave realizes that it might not be possible for him to record all episodes successfully, so instead of focusing on recording all episodes of HackerRank (which may be impossible), he decides to record all consecutively airing episodes whose episode number occurs in some inclusive $[L, R]$ interval such that $R - L$ (i.e., the number of consecutive episodes recorded) is as large as possible.

Given the programming schedule for each season, find L and R episode numbers for largest range of consecutive episodes Dave can record during that season and print these respective values as two space-separated integers on a new line. If two or more such intervals exist, choose the one having the smallest L value.

Input Format

The first line contains a single positive integer, q, denoting number of seasons of HackerRank. The subsequent lines describe each of the q seasons in the following format:

1. The first line contains an integer, n, denoting the number of episodes in the season.
2. Each line i of the n subsequent line contains four space-separated integers describing the respective values of s_{live}, e_{live}, s_{repeat}, and e_{repeat}.

Constraints

- $1 \leq q \leq 100$
- $1 \leq n \leq 100$
- $1 \leq s_l, e_l, s_r, e_r \leq 10^4$
- $s_{live} \leq e_{live}$
- $s_{repeat} \leq e_{repeat}$
- $s_{live} \leq s_{repeat}$

Output Format

On a new line for each season, print two space-separated integers denoting the respective L and R (inclusive) values for the maximum possible range of consecutive episodes Dave can record such that $R - L$ is as large as possible. If more than one such interval exists, choose the interval having the smallest L.

Sample Input

```
3
3
10 20 30 40
20 35 21 35
```

```
14 30 35 50
1
10 20 30 40
3
11 19 31 39
12 38 13 37
10 20 30 40
```

Sample Output

```
1 2
1 1
1 1
```

Explanation

For the first season, Dave records the live airing of episode 1 and the repeat airing of episode 2. Note that it is *not* possible to record episodes 1, 2 and 3 simultaneously.

For the second season, there is only one episode so Dave records from episode 1 to episode 1 and we print `1 1` on a new line.

For the third season, Dave must choose to record either episode 1 or episode 3 (episode 2 starts while episode 1 is still airing and ends after episode 3 starts); he cannot record both, because he only wants to record consecutive episodes. Thus, we pick the episode with the smallest L value, which is episode 1, and print `1 1` as we are only recording episode 1.