Baltic OI '09 P2 - Candy Machine

Time limit: 1.0s **Memory limit:** 512M

Baltic Olympiad in Informatics: 2009 Day 1, Problem 2

In a candy factory, there is a mysterious machine. It produces delicious candies, each a little bit different from the others. The machine has a line of output slots, numbered 1 to n, from which the candies fall out as soon as they are ready. No one really knows how the machine works, but before it starts a production session, it prints a list telling the factory owner, when and from which slot each candy will fall out.

Now, the factory owner can install automatic wagons that move below the output slots to catch the falling candies. Of course, none of the candies should drop on the floor and get spoiled. However, since running the wagons is expensive, the owner would like to install as few wagons as possible.

Write a program that computes the minimum number of wagons needed to catch all candies. Moreover, your program shall output which wagon should catch which candy. The wagons run at a speed of one slot width per second. Before the production process starts, each wagon can be moved to the slot where it should catch its first candy.

Input Specification

The input describes one production session of the machine. The first line contains exactly one integer n, the number of candies produced in that session. Each of the following n lines contain a pair of integers s_i and t_i, the output slot and time for candy i. Each pair (s_i, t_i) is unique.

Output Specification

The first line of output should contain exactly one integer w, the minimum number of wagons needed to catch all candies. The wagons are numbered from 1 to w. The following n lines indicate which wagon will catch which candy. For that purpose, each of these lines contain three integers: the output slot s_j, the output time t_j for some candy j, and a wagon number $w(j)$, such that at time t_j wagon $w(j)$ will be at slot s_j and therefore be able to catch candy j.

Since all candies shall be caught, each slot and time pair given in one input line must appear in exactly one output line (in any order). If there is more than one solution, output any.

Sample Input

```
5
1 1
2 3
1 5
3 4
2 6
```

Sample Output

```
2
1 1 1
2 3 1
1 5 2
3 4 1
2 6 2
```

Constraints

$1 \le n \le 100\,000$

$0 \le s_i, t_i \le 1\,000\,000\,000$

Grading

For test cases worth 20% of the total score, $n \le 85$ and $w \le 4$.

For test cases worth 60% of the total score, $n \le 8\,000$.