

Problem D

The Interval-Merging Problem

Input File: *testdata.in*

Time Limit: 3 seconds

Problem Description

A *close interval* (*interval* for short) is an ordered pair of natural number $[x, y]$, with $x \leq y$. The interval $[x, y]$ represents the set $\{i \in \mathbb{R} \mid x \leq i \leq y\}$. Now, we consider a problem for manipulating intervals. Given a set of closed intervals $\mathcal{I} = \{[a_1, b_1], [a_2, b_2], \dots, [a_k, b_k]\}$, the *Interval-Merging Problem* is to find a minimum-cardinality set of intervals $M(\mathcal{I}) = \{[x_1, y_1], [x_2, y_2], \dots, [x_j, y_j]\}$, $j \leq k$, such that the natural numbers represented by $\mathcal{I} = \bigcup_{i=1}^k [a_i, b_i]$ equal those represented by $M(\mathcal{I}) = \bigcup_{i=1}^j [x_i, y_i]$. Let $|M(\mathcal{I})|$ be the number of the minimum-cardinality set of intervals $M(\mathcal{I})$ on the given interval set \mathcal{I} .

For example, let a set of intervals $\mathcal{I} = \{[1, 5], [2, 6], [3, 6], [8, 10]\}$. The minimum-cardinality set of intervals $M(\mathcal{I}) = \{[1, 6], [8, 10]\}$, and the number of the minimum-cardinality set of intervals $|M(\mathcal{I})| = 2$.

Please write a computer program to count the number of the minimum-cardinality set of intervals $M(\mathcal{I})$ on the given interval set \mathcal{I} .

Technical Specifications

- $1 \leq k \leq 1000$.
- $1 \leq a_i \leq b_i \leq 10000$ for $1 \leq i \leq k$.

Input Format

The first line of the input file contains an integer indicating the number of test cases to follow. For each case, the k intervals are given the following format: The first line contains a positive integer k , denoting the number of the set

of intervals. In the following k lines, each line contains two positive integers separated by a space. The first positive integer indicates the left-endpoint of the interval, and the second positive integer indicates the right-endpoint of the interval.

Output Format

For each test case, output the number of minimum-cardinality set of intervals.

Sample Input

```
2
2
1 4
2 6
4
1 5
2 6
3 6
8 10
```

Sample Output

```
1
2
```