B: Blurred Pictures

Time limit: 2 seconds

Damon loves to take photos of the places he visits during his travels, to put them into frames. All of his photos are in a square format of N x N pixels. He brought back beautiful pictures of the many monuments in Paris, such as the Eiffel Tower or the Louvre, but unfortunately, when he got back home, he realized that all his pictures were blurred on the edges. Looking closely, Damon realizes that he can easily distinguish the blurred pixels from the "good" (i.e., non-blurred) ones and that, luckily, all the non-blurred pixels are connected in such a way that any horizontal or vertical line drawn between two non-blurred pixels goes only through non-blurred pixels. In order to get the best from his failed pictures, he decides to cut out the biggest possible picture without any blurred pixel from each of his photos. And since his frames are all squares, for aesthetic reasons, the cut-out pictures have to be squares too. Damon does not want his pictures to be tilted so he wants the sides of the cut-outs to be parallel to the sides of the original picture.

Important Note

- In the input picture, each row and each column has at least one non-blurred pixel.
- In any two consecutive lines, there are at least two non-blurred pixels in the same column.

Input

The input comprises several lines, each consisting of integers separated with single spaces:

- The first line contains the length N, in pixels, of the input photo;
- Each of the next N lines contains two integers a_i and b_i, the indices of the first (a_i) and the last (b_i) non-blurred pixel on the i-th line.

Limits

- $0 < N \leqslant 100\,000$;
- $0 \leqslant a_i \leqslant b_i < N$.

Output

The output should consist of a single line, whose content is an integer, the length of the largest square composed of non-blurred pixels inside the picture.

Sample Input 1

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

Sample Output 1

```
1
```

Sample Input 2

```
8
2 4
2 4
1 4
0 7
0 3
1 2
1 2
1 1
```

Sample Output 2

```
3
```