ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger

Algorithms Lab

Exercise – *Tetris*

We have a game that can be thought of as a (very) modified tetris. The playing field is w units wide and our goal is to stack blocks on each other on the playing field in such a manner that as many rows as possible are completely filled. Unlike in tetris however, all the building bricks are rectangles with unit height and various (integer) widths. Also unlike in tetris, we are not able to move the bricks around or rotate them, every brick starts on a given (again integer) position and will fall on the top of the top-most brick directly under it. What we can do however is rearrange the order in which the bricks fall. As an added constraint, we want the wall to be robust. This means that no two "cracks" between pairs of bricks should be vertically aligned, even if there are solid bricks between them. Figure 1 shows a bad wall (a) and a good wall (b). Note that this rule obviously doesn't apply for the left and right border of the wall.

Problem You are given the width of the playing field and a set of bricks with their starting and ending locations, ranging from 0 to w. Your goal is to determine the maximum height of a robust wall that can be built using the bricks. In tetris this would be the number of lines which can "disappear". Not all of the bricks available need to be used.

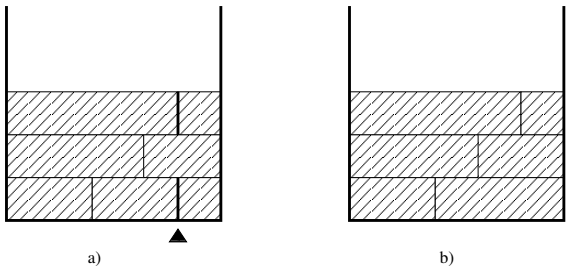

Figure 1: Bad and good wall

Input The first line of the input contains T, the number of test cases. Each of the T subsequent lines will contain one testcase. A testcase starts with w, the width of the playing field and n, the number of bricks. $2n$ numbers follow, numbers $2i$ and $2i+1$ giving the end coordinates of the i-th brick (each coordinate is non-negative and at most w, note that the left end does not necessarily come first). You can expect that all bricks have non-negative width, $0 < w < 500$, and $0 \leq n < 200000$.

Output Your output should contain one line for each testcase. On this line the maximum possible height of solid wall that can be constructed should be output.

Sample Input

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

Sample Output

```
2
1
```