FIT5216: Modelling Discrete Optimization Problems
Inclass Task 15: Pillars and Planks

1 Problem Statement

You are given a number of planks of different lengths and all of unit width, and a number of pillars of different lengths and of different heights. You need to stack the pillars and planks under the following rules

- No plank or pillar can overlap with another
- Each plank must be sitting on a pillar at the left and right extreme ends
- Each pillar must sit on the ground or on a single plank across its entire width
- All objects must fit within the given space

Data for the problem is defined as follows:

```
int: planks;
set of int: PLANK = 1..planks;
array[PLANK] of int: plank_width;
int: pillars;
set of int: PILLAR = 1..pillars;
array[PILLAR] of int: pillar_height;
array[PILLAR] of int: pillar_width;

int: available_width;
int: available_height;
```

The main decision variables are the (x,y) coordinates for each object:

```
set of int: WIDTH = 0..available_width-1;
set of int: HEIGHT = 0..available_height-1;
array[PLANK] of var WIDTH: xk;     % x position of each plank left
array[PLANK] of var HEIGHT: yk;    % y position of each plank
array[PILLAR] of var WIDTH: xr;    % x position of each pillar
array[PILLAR] of var HEIGHT: yr;   % y position of each pillar
```

The aim is to minimize the maximum height of the top of any object.

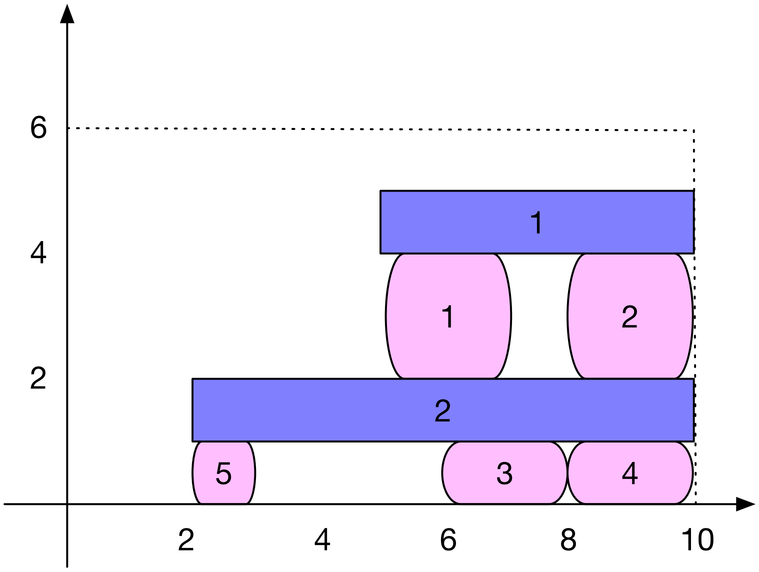
For example given a small data set:

```
planks = 2;
plank_width = [5, 8];
pillars = 5;
pillar_height = [2,2,1,1,1];
pillar_width  = [2,2,2,2,1];
available_width = 10;
available_height = 6;
```

A possible solution is

```
height = 5;
xk = [5, 2];
yk = [4, 1];
xr = [5, 8, 6, 8, 2];
yr = [2, 2, 0, 0, 0];
```

which can be visualized as



Notice how each pillar sits on the ground or entirely on plank 2. Similarly the right and left end of each plank is supported by a pillar. The maximum height of the top of each obstacle is 5 (for plank 1).

2 Instructions

Edit the provided `mzn` model files to solve the problems described above. Your implementations can be tested locally by using the *Run* icon in the MiniZinc IDE or by using,

```
minizinc ./modelname.mzn ./datafile.dzn
```

at the command line.