AI – Constraint programming

Blocks – Fall, 2019 - Gabor

The input for a blocks problem looks like this:

```
Blocks.py 9x18 4x8 7 5 3x11 6 10
```

Adjacent pairs of integers may or may not have an x between them. The first pair of integers is special as it specifies a containing rectangle's height × width that all the other rectangles are to go into.

For all the remaining pairs of integers, they specify the dimensions of rectangles that are to be placed into the containing rectangle. No rectangles may overlap. All other rectangles must fit completely into the first rectangle.

A visual 2D-solution to the above problem would be an all capitalized version of:

```
bBBBBBBcCCCCCCCCCC
BBBBBBBCCCCCCCCCCC
BBBBBBBCCCCCCCCCCC
BBBBBBB dDDDDDDDDD
BBBBBBB DDDDDDDDDD
aAAAAAAADDDDDDDDDD
AAAAAAAADDDDDDDDDD
AAAAAAAADDDDDDDDDD
AAAAAAAADDDDDDDDDD
```

Notice that in this case there is a hole in the center. Our way of specifying the output in a 1D form will be to specify the dimension of each new rectangle as it is encountered scanning from left to right, top to bottom. Dimensions are specified as height × width (orientation matters in the output), and holes are specified as rectangles, too. The above rectangles would be specified as:

```
Decomposition: 5x7 3x11 2x1 6x10 4x8
```

The point of first encounter of each rectangle is indicated by the corresponding lowercase letter (and it was for this illustrative purpose that they weren't all capitalized in the above display). The letters were chosen to correspond to the order in which the rectangles were listed in the input. Note that the hole is explicitly specified. If there were non-contiguous "rectangles" of holes, then each hole rectangle would be specified separately. The grader will look for the first occurrence of the word `Decomposition` followed on the same line by at least as many integers as (twice the number of rectangles in the input) less 2. The integers may or may not be separated by x's or spaces.

Some problems will have no solution.
The output in this case should be one of (take your pick):

```
No solution
Impossible to solve
No decomposition possible
```

Here are two problems with no solution:

```
10 13 3x6 2 5 4x10 7 9 1x1
8x5 4x2 4 2 2x4 2 4 4 2
```

Here are three additional starter test cases – can you find solutions for them?

```
4x7 7x4
2 3 1 2 2 2
4 8 4x1 1x6 1 3 3x1 1x3 1 3 6x1 1x4
```