NAVIGATION

B4B36ZUI
tasks
Task 1 - EN - Path Planning
Task 1 - CS - Path Planning
Task 2 - EN - Nonograms
Task 2 - CS - Malované křížovky
Task 3 - EN - Monkey and banana
Task 4 - EN - Grid-world
Přednášky / Lectures 2018
Cvičení / Labs 2018
Introduction to Artificial Intelligence (B4B36ZUI,BE4B36ZUI)

ALL COURSES

Summer 2017 / 2018
Older

# Task 2 - EN - Nonograms

Table of Contents –

## Assignment

Your task is to write an algorithm that solves a nonogram. The nonograms are brain teasers, defined by a legend linked to a grid used to draw a picture. Each number in the legend sets the number of the following boxes filled with color which belongs to the given number. The following rules hold:

- there is always at least one empty box between two blocks of the boxes filled with the same color
- there does not have to be an empty box between the boxes filled with different color
- the order of the numbers in the legend corresponds to the order of the blocks of the boxes (from left to right, from top to bottom)

## How to solve the task

### Input

The program is going to read the file from the standard input that has the following format:

```
NUMBER_OF_ROWS,NUMBER_OF_COLUMNS
CONSTRAINTS_FOR_ROW_1
CONSTRAINTS_FOR_ROW_2
...
CONSTRAINTS_FOR_ROW_M
CONSTRAINTS_FOR_COLUMN_1
CONSTRAINTS_FOR_COLUMN_2
...
CONSTRAINTS_FOR_COLUMN_N
```

where each constraint has the format:

```
COLOR_1,NUMBER_1,COLOR_2,NUMBER_2,...,NUMBER_K
```

where the field “color” applies to the following number and is in a format of the character that represents the color you will use for drawing (for example '#'). The field “number” sets the size of the given block.

Input examples: csp_example.txt, input.txt, krtkek.txt, dino.txt

Output examples: csp_example.txt.out input.txt.out krtek.txt.out, dino.out.txt

### Output

- Use the standard output for the picture.
- Draw the picture line-by-line, for the colored box write the sign of the given color, for the empty box write “_”.
- In case multiple pictures satisfy the constraints, draw all of them separated with an empty line.
- If the solution does not exist, output “null”.

### Important Information

Think thoroughly through the way how you formalize the task - what is going to be a variable, how are the constraints and whether the edge consistency algorithms may be applied. Start with easy depth search and improve it by using CSP techniques.

In your main class write to the comment section what formalisation you have chosen, how you represent the constraints and what CSP algorithms are implemented.

### Score

You can get up to 14 points for your task - you will get all points if you find all solutions of the problem using CSP techniques in combination with search and using the arc consistency algorithm (AC-3).

**It is more important to have a correct CSP implementation and integration with AC-3 rather than the fastest algorithm for solving nonograms.**

## Implementation details

- Create a package named “student”
- Create the main class CSPMain in the “student” package, which is going to implement the *main* method.
- Your algorithm has to print out **all solutions**
- Submit the zipped package structure containing all your classes.

## Deadline

Week 5 - 08/04/2018 23:59