Modeling Discrete Optimization Assignment: **Gang Warfare**

1 Problem Statement

The principal of Headley Heights has a problem. The school chapel has been vandalized repeatedly over the past few weeks by a gang mischievous students who **must** be brought to justice! The culprits can be identified by interviewing each student for a confession to the crime. However, members of the same gang will never snitch on each other. Hence, the order that each student is interviewed is critically important to disrupting the gang's loyalty and maximizing the quality of the confessions. In this assignment you will help the principle find an order of the student interviews to maximize confession quality and ultimately determine which gang is responsible for vandalizing the school chapel.

The school has t students organized into n different gangs. The value m_i $(1 \leq i \leq n)$ denotes the number of students in each gang and the students are numbered sequentially $1..t$ in those gangs. For example, $n = 4, m = [3, 2, 2, 3]$ results in the following gangs of students,

$m_1 = \{1, 2, 3\}$
$m_2 = \{4, 5\}$
$m_3 = \{6, 7\}$
$m_4 = \{8, 9, 10\}$

Notice that $t = \sum_{i=1}^{n} m_i$. The first member (i.e. the lowest student number) is the **leader** of each gang. In the example above,

the leader of m_1 is student 1
the leader of m_2 is student 4
the leader of m_3 is student 6
the leader of m_4 is student 8

1.1 Stage A

The principal has to interview each gang member at one of t time slots (i.e. one time slot for each student). No two members of the same gang can be interviewed one after the other, as this would allow collusion as they met entering and leaving the office.

The success of the confession is measured in confession points. A non-leader member interviewed before their gang leader gives 1 point, if interviewed after their leader they see his/her bravado and give 0 points. For stage A the principle gains 3 confession points if a gang leader is interviewed in the morning, at time $< t/2$, when the principal is fresh and 0 points otherwise. The aim is to order the interviews to maximize the total number of confession points.

1.2 Stage B

In reality the gang rivalries are very important in understanding what is happening in the school. There are currently r rivalries going on where $g1_i, g2_i$ $(1 \leq i \leq r)$ indicate a pair of gangs involved in a rivalry. For Stage B, no two members of gangs in a rivalry can be interviewed one after the other, as this would result in violence as they met entering and leaving the office.

Interviewing the leader of a gang (e.g. gang$_1$) after all the members of another gang (e.g. gang$_2$) in a rivalry with gang$_1$ makes the leader of gang$_1$ feel that the principal is working against them and the leader confesses $3+2l$ points where l is the number of other members of their own gang (i.e. m_1) not interviewed yet. In all other cases, interviewing a gang leader gives 3 points if interviewed in the morning, at time $< t/2$, when the principal is fresh and 0 points otherwise. As in Stage A, any non-leader member interviewed before their gang leader gives 1 point, if interviewed after their leader they see his/her bravado and give 0 points. In this stage, the aim is still to order the interviews to maximize the total number of confession points.

2 Data Format Specification

The input data is given as a file `data/gw_*.dzn` in MiniZinc data format:

`n =` n`;`
`m = [`$m_1, \ldots, m_n$`];`
`r =` r`;`
`g1 = [`$g1_1, \ldots, g1_r$`];`
`g2 = [`$g2_1, \ldots, g2_r$`];`

The solution should be given as one array of size t and the objective function value,

`order = [` t numbers, being the person interviewed at time t `];`
`obj = value;`

Note that data files for Stage A will always have no rivalries, though they still will include the parameters r, $g1$ and $g2$ with definitions

r = 0; g1 = []; g2 = [];

An example of a stage B problem is when there are four gangs with [3,2,2,3] members each and two rivalries between gangs 1 and 3 and gangs 2 and 4 the data file is,

```
n = 4;
m = [3,2,2,3];
r = 2;
g1 = [1,2];
g2 = [3,4];
```

then one solution is

```
order = [7, 5, 6, 4, 1, 8, 3, 10, 2, 9];
obj = 22;
```

which gains 22 confession points.

Hint: The default search of Gecode may not solve these problems very well, so you should consider adding a programmed search strategy, which might be quite different for Stage A and Stage B.

3 Instructions

Edit `gangwarfare.mzn` to solve the optimization problem described above. Your `gangwarfare.mzn` implementation can be tested on the data files provided. In the MiniZincIDE use the *play* icon to test your model locally. At the command line use,

```
mzn-gecode ./gangwarfare.mzn ./data/<inputFileName>
```

to test locally. In both cases, your model is compiled with MiniZinc and then solved with the GECODE solver.

Resources You will find several problem instances in the `data` directory provided with the handout.

Handin From the MiniZinc IDE, the *coursera* icon can be used to submit assignment for grading. From the command line, `submit.py` is used for submission. In both cases, follow the instructions to apply your MiniZinc model(s) on the various assignment parts. You can submit multiple times and your grade will be the best of all submissions.[1] It may take several minutes before your assignment is graded; please be patient. You can track the status of your submission on the *programming assignments* section of the course website.

4 Technical Requirements

For completing the assignment you will need MiniZinc 2.0.x and the GECODE 4.4.x solver. Both of these are included in the bundled version of the MiniZinc IDE 0.9.9 (`http://www.minizinc.org`). To submit the assignment from the command line, you will need to have Python 2.7.x installed.

[1] Problem submissions can be graded an unlimited number of times. However, there is a limit on grading of **model submissions**.