

离散优化建模：作业九

寿辰阅兵

1 问题描述

巨鳌十分感谢女娲让他成为龙宫的丞相。于是他安排他手下最精锐的虾兵蟹将（下文全部称作“士兵”）来庆祝女娲万岁寿辰。在最初的阵型中，士兵排成很多列，而队长在最前面。然后所有士兵会出列然后依次走过女娲跟前，整个队伍由队长带头。每一个士兵只能在他所在列前面所有士兵出列之后才能出列。

女娲认为一个整齐的队列应该是所有士兵有相同的身高。所以她会在阅兵过程中都很开心直到她能明显察觉到连续两个士兵的身高有明显差异。我们的任务是决定一个士兵出列走过女娲跟前的顺序来让女娲尽可能长时间保持满意。

2 数据格式说明

寿辰阅兵问题的输入是一个名为 `data/paradep.dzn` 的文件，其中 p 是问题的序号。数据文件定义了以下数量：

- n 是队列中士兵的序号，
- $height$ 是包含每一个士兵身高的数组，
- $cols$ 他们排列的阵列中列的数量，
- $delta$ 是女娲年老而不好的视力下难以察觉身高的差距。

问题的目标是安排士兵的出列顺序 pos 。

队长的序号永远是 1，士兵 2 号是第一列的第一个士兵，士兵 $cols + 1$ 是最后一列的第一个士兵，士兵 $cols + 2$ 是第一列的第二个士兵，如此类推。

数据声明和主要的决策变量如下：

```
int: n;                                % number of soldiers
set of int: SOLDIER = 1..n;
set of int: POS = 1..n;
array[SOLDIER] of int: height; % height of each soldier
int: cols;                             % number of columns of soldiers
int: delta;                             % height delta allowed
```

```
array[SOLDIER] of var POS: pos; % order position of each soldier
```

每一阶段的输出需要的形式如下：

```
pos = [士兵出列顺序];  
obj = 女娲保持满意的时间;
```

女娲在检阅过程中对队列感到满意，直到队列中出现第一个与其之后士兵身高明显不同（差距大于 δ ）的士兵出列。如果这没有发生，则女娲对整个队列都满意。（ $obj = n$ ）

这次作业的目的是对这个问题进行建模，然后给出一个基本的搜索策略（只需要一个 `int_search` 注解）可以令到女娲尽可能长时间高兴。要注意到这个是一个排列问题，所以可以使用不同的视角来建模。

比如对于以下数据文件

```
n = 12;  
height = [4, 3, 3, 5, 2, 5, 3, 4, 2, 1, 2, 4];  
cols = 4;  
 $\delta$  = 1;
```

士兵原本以四列排列在队长后面，阵型如下

			1	
2	3		4	5
6	7		8	9
10	11		12	

一个最优解就可以是

```
obj = 11;  
pos = [1, 9, 4, 2, 5, 11, 6, 3, 7, 12, 8, 10];
```

注意到士兵 11 在前面的两个士兵都出列后，他排第 8 个出列。这是在士兵 3（排第 4）和士兵 7（排第 6）出列之后。你也可以注意到队列中士兵的身高是 4, 5, 4, 3, 2, 3, 2, 2, 3, 4, 5, 1，而女娲直到倒数第二个士兵之前都是满意的。

3 指引

你可以编辑 `parade.mzn` 模型文件来解决上述优化问题。你实现的模型 `parade.mzn` 可以用提供的数据文件进行测试。在 MINIZINC IDE 中，你可以通过点击 *Run* 按钮在本地测试和运行。或者在命令行中输入

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

进行本地测试和运行。两种情况下，你的模型都是用 MINIZINC 进行编译然后用 GECODE 求解器求解。

参考资料 你可以在 `data` 文件夹下找到讲义中的几个问题实例（的数据文件）。

提交作业 这次的作业包含有 3 个答案提交部分和 5 个模型提交部分。对于答案提交部分，我们将会提交求解器求解你的模型所得到的最好/最后的答案，然后检查它的正确性和质量。对于模型提交部分，我们将会提交你的模型文件 (.mzn) 然后用一些未公开的数据文件来做进一步检查。

在 MINIZINC IDE，点击 *Submit to Coursera* 图标可以用于提交作业。若采用命令行方式，`submit.py` 可以用于提交作业。无论采用那种方法，你都需要根据本指引中的要求完成作业各部分的 MiniZinc 模型。你可以多次提交，最终作业分数是你的最高的一次。¹作业的打分过程可能需要几分钟，请耐心等待。你可以在课程网站上的 *编程作业* 版块查看你的作业提交状况。

4 软件要求

为了完成作业，你需要安装 MINIZINC 2.1.x 和 GECODE 5.0.x 求解器。这些软件都会包含在 MINIZINC IDE 2.1.X (<http://www.minizinc.org>) 的集成版本中。如果你需要通过命令行提交作业，你需要安装 Python 3.5.x。

¹问题解的提交并没有次数限制。但是，**模型提交部分**只能提交有限次。