

FIT5216: Modelling Discrete Optimization Problems

Inclass Task 9: Reindeer

1 Problem Statement

You have 4 reindeer `Lancer`, `Quentin`, `Ezekiel` and `Rudy`. You fly your sled by tying them in a single line.

- `Lancer` cant be next to `Ezekiel`
- `Rudy` has to be ahead of `Quentin` or ahead of `Lancer`

How many ways can you arrange your Reindeer?

Build a MiniZinc model `reindeer.mzn` to find out. Use the variable declarations and output statement:

```
set of int: POS = 1..4;
enum REINDEER = { Lancer, Quentin, Ezekiel, Rudy };
array[REINDEER] of var POS: x;
output ["x = array1d(REINDEER,\(x));\n"];
```

You can collect all solutions using `-a` from the command line, or in the IDE under solver configuration, checking “User-defined behavior” and setting stop after 0 solutions.

Model the problem in an alternate fashion.

Build a MiniZinc model `reindeer_inv.mzn` to find out. Use the variable declarations and output statement:

```
set of int: POS = 1..4;
enum REINDEER = { Lancer, Quentin, Ezekiel, Rudy };
array[POS] of var REINDEER: y;
output ["y = \(y);\n"];
```

Which is easier?

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
```

at the command line.