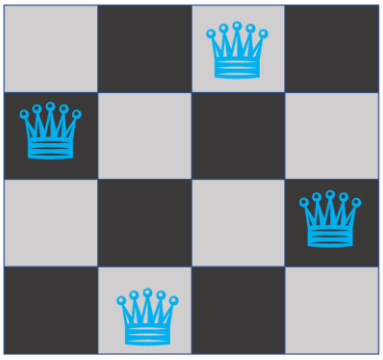
CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computing@CIT

DECISION ANALYTICS.

Lab06: The N-Queens problem



BACKGROUND.

Is it possible to place N queens on a chess board of $N \times N$ fields so that no two queens are threatening each other, i.e. are not in the same line horizontally, vertically, or diagonally.

Task 1.

Write a Python program that creates a CP-SAT model of the n-queens problem. Choose a suitable model and add all variables and constraints to the model.

Task 2.

Implement a `CpSolverSolutionCallback` that prints each solution as follows:

```
---------------------------------
| Q |   |   |   |   |   |   |   |
---------------------------------
|   |   |   |   |   |   | Q |   |
---------------------------------
|   |   |   |   | Q |   |   |   |
---------------------------------
|   |   |   |   |   |   |   | Q |
---------------------------------
|   | Q |   |   |   |   |   |   |
---------------------------------
|   |   |   | Q |   |   |   |   |
---------------------------------
|   |   |   |   |   | Q |   |   |
---------------------------------
|   |   | Q |   |   |   |   |   |
---------------------------------
```

Task 3.

Solve the model and evaluate the performance for finding 1 solution (use `self.StopSearch()` in the `OnSolutionCallback` function) and for finding all solutions using the `solver.ResponseStats()` function.