

1 E.A.5.2

1.1 n -Queens

Il problema n -Queens si potrebbe codificare come problema CSP nel seguente modo. Siano (X, D, C) t.c.

- $X = \{Q_1, \dots, Q_n\}$
- $D = \{D_1, \dots, D_n \mid D_i = \{1, \dots, n\}\}$
- vincoli (due regine non possono stare sulla stessa riga e non si possono attaccare in diagonale):
 - $\text{alldiff}(Q_1, \dots, Q_n)$
 - $\langle \{Q_i, Q_j\}, \Delta \rangle$ t.c.
 - $i \neq j \wedge \Delta = \{(r_i, r_j) \mid r_i, r_j = 1, \dots, n \wedge |i - j| \neq |r_i - r_j|\}$

1.2 5-Queens

- $X = \{Q_1, Q_2, Q_3, Q_4, Q_5\}$
- $D = \{D_1, D_2, D_3, D_4, D_5 \mid D_i = \{1, 2, 3, 4, 5\}\}$
- vincoli:
 - $\text{alldiff}(Q_1, Q_2, Q_3, Q_4, Q_5)$
 - per le diagonali vedere l'ultima pagina...

1.3 MiniZinc code

```
include "alldifferent.mzn";

int: n = 8; /* l'editor va in crash a n = 21 */
array[1..n] of var 1..n: queens;
constraint alldifferent(queens);
constraint forall(i in 1..n, j in (i + 1)..n)(
    abs(queens[j] - queens[i]) ≠ j - i
);
```

Piccola nota: effettivamente il vincolo descritto in MiniZinc non è identico alla definizione in notazione matematica, andrebbe riscritto così:

```
include "alldifferent.mzn";

int: n = 8; /* l'editor va in crash a n = 21 */
array[1..n] of var 1..n: queens;
constraint alldifferent(queens);
constraint forall(i in 1..n, j in 1..n)(
    i ≠ j → abs(queens[j] - queens[i]) ≠ abs(j - i)
);
```

Con $n = 20$ l'esecuzione del primo codice dura circa 821ms, mentre l'esecuzione per la seconda versione dura circa 1s e 516ms.

I vincoli per cui le regine non si possono attaccare in diagonale

[illegible]