

Programmation par contraintes en Oz

Exercice 1

Une *séquence magique* de longueur n est une séquence d'entiers x_1, \dots, x_n entre 0 et $n - 1$ telle que pour tout $i = 0, \dots, n - 1$ le nombre i paraît exactement x_{i+1} fois dans la séquence.

Un exemple d'une séquence magique de longueur 7 est

3, 2, 1, 1, 0, 0, 0

car 0 paraît $x_1 = 3$ fois dans la séquence, 1 paraît $x_2 = 2$ fois dans la séquence, ..., 6 paraît $x_7 = 0$ fois dans la séquence.

Écrire une fonction en Oz, en utilisant les contraintes à domaines finis, qui prend un entier n en argument et qui envoie la liste de toutes les séquences magiques de longueur n .

Voici les résultats attendu de votre fonction pour quelques valeurs de n :

n	résultat
1	nil
2	nil
3	nil
4	[[1 2 1 0] [2 0 2 0]]
5	[[2 1 2 0 0]]
6	nil
7	[[3 2 1 1 0 0 0]]

Conseil : Utiliser le propagateur FD.exactly.

Exercice 2

Un *triangle de distances* de taille n est un triangle de largeur n et profondeur n tel que ses positions sont occupées par des valeurs différentes entre 1 et $\frac{n*(n+1)}{2}$, et tel que toute valeur (sauf sur la première ligne) est la distance absolue entre les deux valeurs qui se trouvent sur la ligne au-dessus, directement à la gauche et directement à la droite. Par exemple, un triangle de distance de taille 5 possible est le suivant :

```
      6      14      15      3      13
       8       1       12      10
        7       11       2
         4        9
          5
```

Puisque $8 = |6 - 14|$, $1 = |14 - 15|$, $12 = |15 - 3|$, ..., $5 = |4 - 9|$.

1. Écrire un programme en Oz, en utilisant les contraintes de domaine fini, qui génère tous les triangles de distance de taille 5.

2. Écrire une procédure en Oz, en utilisant les contraintes de domaine fini, qui prend en entrée un entier n et qui génère tous les triangles de distance de taille n . Tester au moins avec les valeurs pour n entre 1 et 7.

Indication : Créer d'abord un triangle de variables à domaine fini comme une liste qui contient des listes de variables à domaine fini. Servez vous des fonctions de la bibliothèque standard :

- `{List.make N}` envoie une liste avec N variables fraîches
- `{FD.list N spec}`, où `spec` est la spécification d'un domaine, envoie une liste avec N variables qui ont toutes le domaine `spec`.

N'oubliez pas que `FD.distribute` attend un vecteur avec des variables au niveau 1, c-à-d il va falloir aplatir la structure créée avant de la passer à `FD.distribute`.