

---

## CC1 : mini-projet de modélisation en MiniZinc

---

Ce projet est à réaliser individuellement. Vous mettrez les sources sur gitlab, en indiquant votre nom dans le nom du projet et en m'ajoutant comment revieweuse (mon identifiant est : sara.tari). Je dois avoir accès aux projets au plus tard le dimanche 6 décembre 2020 à 23h59, au-delà de ce moment les modifications de vos projets ne seront plus acceptées.

Vous devrez commenter votre modèle pour expliquer votre raisonnement, si vous n'arrivez pas à mettre en place les contraintes, expliquer les difficultés rencontrées et votre début de raisonnement.

Votre projet consiste à écrire un modèle MiniZinc pour résoudre l'énigme suivante. On dispose de  $N$  seaux d'eau dont on connaît la capacité respective. Partant d'une situation initiale dans laquelle on connaît le remplissage de chacun des seaux, il faut arriver à une situation finale donnée au moyen d'une série de transfert d'eau entre les seaux. Chaque transfert d'eau s'effectue sans pertes et consiste à vider un seau  $X$  pour remplir un seau  $Y$ . Un transfert de  $X$  à  $Y$  est impossible si  $X$  est vide ou  $Y$  est rempli intégralement. On ne dispose pas d'instruments de mesure.

Exemple : On dispose de trois seaux d'eau, A, B, C respectivement de capacités de 8 litres, 5 litres et 3 litres. Initialement, A est rempli intégralement alors que B et C sont vides. Combien de transferts d'eau doit on réaliser au minimum pour que A et B contiennent chacun 4 litres et que C soit vide ?

Le modèle écrit correspondra à la forme généralisée du problème. Pour le tester, vous créerez au moins trois fichiers de données distincts (.dzn). Chaque fichier doit contenir les valeurs des paramètres suivants :

- le nombre de seaux (dans l'exemple : 3),
- un tableau contenant la capacité de chaque seau (dans l'exemple :  $[8, 5, 3]$ ),
- un tableau correspondant à l'état initial (dans l'exemple :  $[8, 0, 0]$ ),
- un tableau correspondant à l'état final (dans l'exemple :  $[4, 4, 0]$ ),
- un nombre d'étapes maximum (une étape correspond à un unique transfert).

Dans votre modèle, vous aurez besoin de 4 variables de décision, dont un tableau permettant d'indiquer quel seau est vidé de l'état  $i - 1$  à  $i$  (cette information sera stockée dans la case d'indice  $i$  et un tableau indiquant le seau rempli à chaque étape). Une des autres variables est un tableau indiquant la quantité d'eau dans chaque seau à chaque état, ce tableau sera rempli uniquement à partir des variables d'action évoquées dans la phrase précédente, sauf pour l'état initial et final. Pour ce tableau vous ferez attention au domaine des variables.

Il sera primordial d'interdire les transferts inverses sur des étapes consécutives (contrainte).

Pour que le modèle fonctionne, il faudra mettre une priorité sur le choix des seaux à vider et remplir avec une *stratégie* (technique de modélisation avancée). Au moment d'indiquer la résolution par le solveur du problème, la commande suivante (attention à ajuster en fonction du nom de vos variables et paramètres) permettra d'indiquer qu'on vide un seau avant d'en remplir un autre :

```
solve : : int_search([if (i+1) mod 2 == 0 then vide[(i+1) div 2] else rempli[(i+1) div 2]  
endif | i in 1..2*etapes],input_order,indomain_min,complete) minimize <var à minimiser> ;
```

Lorsque le modèle sera résolu, vous afficherez la quantité de chaque seau à chaque étape.