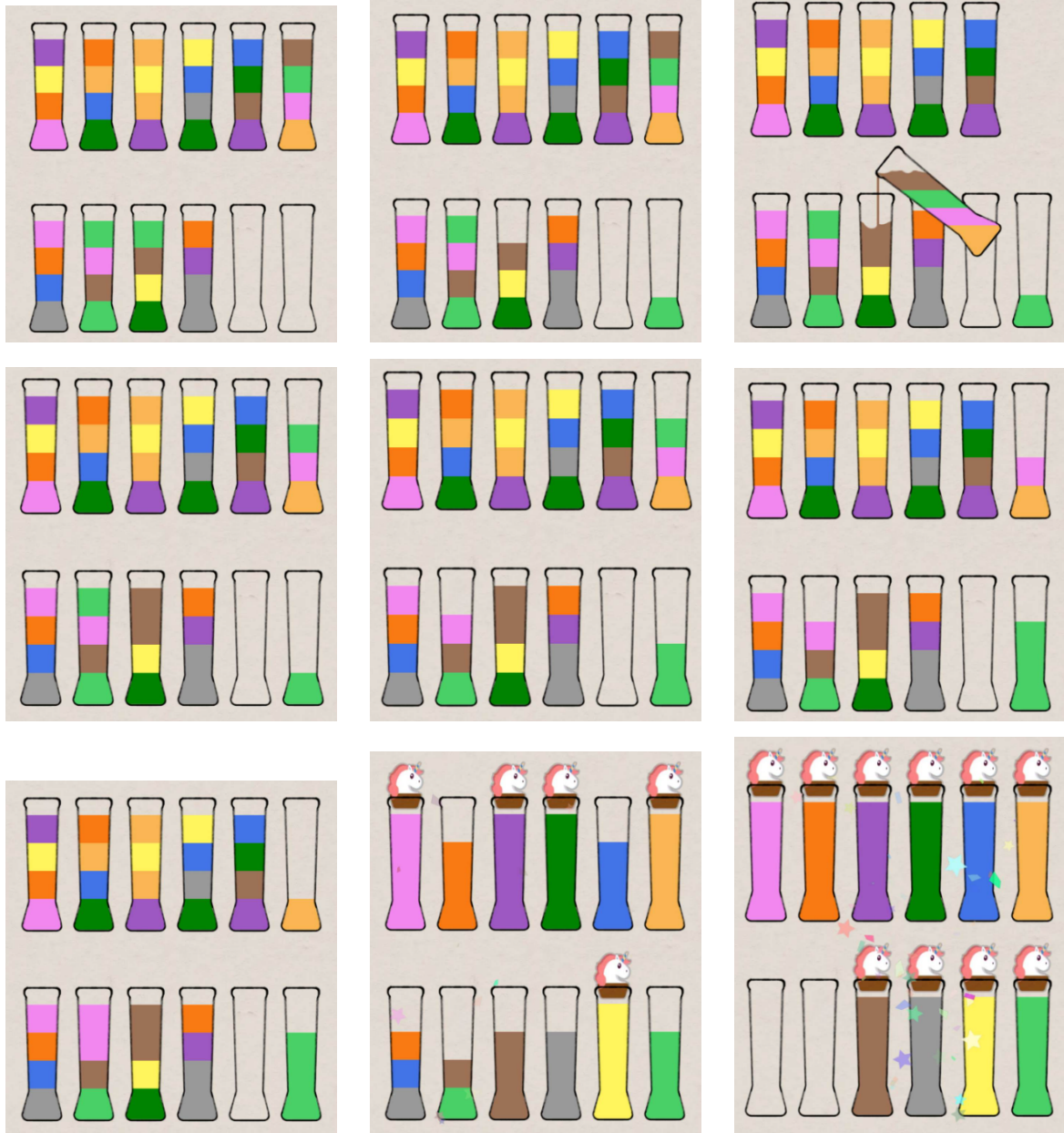


Florian Bourse

# Water sorting puzzle



Généralisons ce jeu en deux problèmes : un problème de décision et un problème d'optimisation :

#### WaterSort :

**Instance :**  $n$  piles de capacité  $c$ .

**Solution :**  $V$  si il existe une suite de coup permettant d'arriver dans un état où chaque pile est soit remplie avec une seule couleur, soit vide.  $F$  sinon.

#### WaterSort\* :

**Instance :** Une instance de WaterSort dont la solution est  $V$ .

**Solution :** Une suite de coup permettant d'arriver dans un état où chaque pile est soit remplie avec une seule couleur, soit vide. **Optimisation :** Minimiser le nombre de coups.

Les deux sections suivantes peuvent être traitées de manière indépendantes.

## 1 Difficulté du problème

Étudier la difficulté du problème de décision WaterSort. On pourra utiliser le résultat suivant : le problème 3-Partition défini ci-dessous est NP-difficile, même si les entrées sont données en unaire.

#### 3-Partition :

**Instance :** Un ensemble de  $3m$  entiers  $a_1, \dots, a_{3m}$ , tels que  $\sum_{j=1}^{3m} a_j = mB$ .

**Solution :**  $V$  si il existe une partition de  $\llbracket 1; 3m \rrbracket$  en  $m$  triplets  $(j_{i,1}, j_{i,2}, j_{i,3})_{i=1\dots m}$  tels que  $a_{j_{i,1}} + a_{j_{i,2}} + a_{j_{i,3}} = B$ .  $F$  sinon.

## 2 Résolution du problème

Donner un algorithme qui permet de déterminer en inspectant une fois chaque pile un minorant du nombre de coups requis pour arriver dans un état où chaque pile est soit remplie avec une seule couleur, soit vide.

En déduire un algorithme permettant de résoudre WaterSort\* et l'implémenter.