

# Modélisation : chaines binaires

Master 1 informatique I2L / WeDSci  
2024 / 2025

## Exercice 1 : Sac à dos (Knapsack)

Pour partir en voyage, un voyageur remplit son sac à dos d'objets. Chaque objet a un *profit* évalué par le voyageur en fonction de son intérêt et un *poids*. Naturellement, un poids limite est imposé par le transporteur. Le but de ce problème est de remplir le sac d'objets afin d'avoir le plus grand profit tout en respectant la capacité maximale.

Questions :

- a - Donner un exemple d'un tel problème
- b - Modéliser le problème en définissant l'espace de recherche, l'objectif à maximiser et la contrainte à respecter.
- c - Définir une fonction de fitness qui combine l'objectif et traduit la contrainte par une pénalité.

## Exercice 2 : Fonction d'évaluation

Le but est de coder une fonction d'évaluation (aussi appelé fonction objectif) du problème du sac à dos.

Questions :

- a - Coder une classe `Solution` ayant pour attribut un vecteur de booléen et un réel nommé `fitness` représentant la qualité du sac à dos. Vous pouvez ajouter une méthode d'affichage de cette solution.
- b - Coder une classe `Knapsack` qui permet de calculer la fonction d'évaluation du problème de sac à dos. Cette classe doit avoir un constructeur qui permet de lire le fichier d'instance du problème, et une méthode qui permet de mettre à jour l'attribut `fitness` en fonction du vecteur de booléens d'une `Solution`.
- c - Tester votre classe.

## Exercice 3 : Recherche aléatoire

Le but est de coder une recherche aléatoire qui consiste à évaluer  $n$  solutions aléatoires, et d'en retenir la meilleure.

Questions :

- a - Coder la recherche aléatoire. Le nombre d'évaluation de la recherche doit être un paramètre de l'algorithme. Comme nous allons coder plusieurs algorithmes de recherche, vous pouvez mettre en place une classe générique **Search**, et une classe **RandomSearch** qui dérive de la classe **Search**.
- b - Tester la recherche en exécutant au moins 100 fois l'algorithme et enregistrer la performance de la recherche aléatoire dans un fichier avec deux colonnes : nombre d'évaluation et performance.