

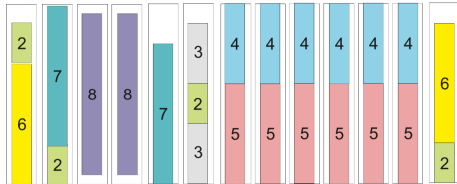
# Projet – Bin Packing 1D

Ce projet est à réaliser par groupe de **2 étudiants maximum**.

Date limite de rendu : **lundi 07 juin à 23h59** au plus tard.

Règle : **5 points de moins par jour de retard**.

L'objectif est d'implémenter des algorithmes de résolution du problème de Bin Packing à une dimension.



On dispose de  $n$  items ( $I=\{1, \dots, n\}$ ) de taille monodimensionnelle ( $t_i$  : taille de l'item  $i$ ) et d'un nombre illimité de boîtes, appelées « bins », de taille monodimensionnelle identique  $t$ . L'objectif est

de ranger tous les items dans des bins en utilisant le moins de bins possible. Il faut donc minimiser le nombre de bins utilisés.

Vous trouverez sur Claroline différents jeux de données sur lesquels tester vos algorithmes.

Vous devez :

1. Pour chaque jeu de données, déterminer une borne inférieure du nombre de bins à utiliser.
2. Implémenter un algorithme, généralement appelé « FirstFitDecreasing », qui ordonne les items dans un ordre décroissant de leur taille et qui essaye de les insérer dans cet ordre dans les bins déjà présent du premier au dernier et qui ajoute un bin si l'item ne tient dans aucun bin existant. A appliquer sur tous jeux de données.
3. En utilisant un package de programmation linéaire :
  - a. Trouver la solution optimale du problème « binpack1d\_00.txt ».
  - b. Essayer de déterminer à partir de quelle limite (nombre d'items) il devient difficile d'obtenir une solution. Pour cela construisez vous-même des jeux de données de plus en gros en dupliquant par exemple les items du problème « binpack1d\_00.txt ».
4. Créer au moins 2 générateurs aléatoires de solutions :
  - a. Un, créant autant de bins que d'items et mettant un item par bin.
  - b. Un, utilisant la procédure « FirstFit » précédente mais en prenant en entrée un ordre aléatoire au lieu d'ordonner les items dans l'ordre décroissant de leur taille.
5. Créer des opérateurs de voisinage, dont au moins :
  - a. Déplacer un item d'un bin vers un autre
  - b. Échanger deux items de deux bins différents
6. Implémenter l'algorithme du « Recuit simulé » et le tester sur tous les jeux de données. Vous devez décrire et illustrer l'analyse des paramètres utilisés.
7. Implémenter l'algorithme « Tabu Search » et le tester sur tous les jeux de données. Vous devez décrire et illustrer l'analyse des paramètres utilisés.
8. Comparer l'ensemble des algorithmes et discuter les résultats.

Vous devez fournir un rapport en PDF expliquant et illustrant l'ensemble du travail réalisé, et fournir le code associé (en indiquant comment l'exécuter). Tout ceci devra être déposé dans un ZIP à votre nom dans la « Zone de dépôt » du module Claroline associé au cours.