

Algorithmique et complexité

TD 6/7 – Problème d’empaquetage

Exercices complémentaires

Remarque: Les éléments de correction fournis dans le sujet ne sont pas forcément complets. Ils sont là pour vous guider dans votre travail personnel. Nous vous invitons à rédiger une solution comme vous le feriez à l’examen et, si vous avez des questions, à vous tourner vers votre chargé de TD.

Exercice 1 : Temps limité

L’entreprise ne souhaite plus louer de serveurs à l’extérieur : elle dispose donc uniquement de M serveurs. Chaque serveur est disponible B minutes par jour mais l’objectif de l’entreprise est maintenant de réduire le plus possible le temps d’utilisation de ses serveurs. Elle souhaite donc écrire un programme qui lui permet de déterminer s’il est possible de traiter les demandes en un temps $T \leq B$ fixé.

Question 1

Donner une définition formelle de ce nouveau problème de décision.

Éléments de correction :

BIN PACKING

Entrée :

- un ensemble O de N objets de taille $o_i \in \mathbb{N}$
- $T \in \mathbb{N}$ une taille à ne pas dépasser
- M un nombre de sacs disponibles

Question : Existe-t-il une affectation des N objets dans les M sacs, telle que la somme des tailles des éléments de chaque sac est inférieure ou égale à T .

Le problème de décision est le même sauf que T remplace B !

Question 2

Quel est le problème d’optimisation que l’entreprise veut résoudre ?

Éléments de correction :

BIN PACKING

Entrée :

- un ensemble O de N objets de taille $o_i \in \mathbb{N}$
- M un nombre de sacs disponibles

Question : Quelle est la **plus petite valeur** de T telle qu’il existe une affectation $(f : [1, N] \rightarrow [1, M])$ vérifiant $\forall j \in \text{img}(f). \sum_{i \in f^{-1}(\{j\})} o_i \leq T$.

Question 3

Proposez un algorithme glouton pour résoudre ce problème.

Éléments de correction :

On peut, par exemple, placer l’objet dans le sac ayant systématiquement la plus grande place restante (une sorte de Worst-fit). **Écrivez le code Python de cette algorithme et testez le.**