L3-INFO Algorithmique et Modélisation

Quick – 25 mars 2022 – durée 1 h

Sont interdits : les documents, les ordinateurs, les téléphones (incluant smartphone, tablettes,... tout ce qui contient un dispositif électronique).
Seuls les dictionnaires papier pour les personnes de langue étrangère sont autorisés.
Il sera tenu compte de la qualité de la rédaction et de la clarté de la présentation (2 pts).
En cas d'incompréhension du sujet, préciser les hypothèses de travail que vous faites et continuer.
Le barème **indicatif** : Exercice 1 : 12 pts ($35 \sim mn$) ; Exercice 2 : 6 pts ($20 \sim mn$), Relecture : ($5 \sim mn$)
Les 2 exercices sont indépendants.

I : Randonnée

Que fait un grenoblois lorsqu'il ne fait pas de l'Informatique ? De la montagne évidemment ! Mais il doit faire attention à son paquetage, pas trop lourd mais d'intérêt maximal. Informellement, son problème est de sélectionner des objets parmi un ensemble donné d'objets avec une *valeur* maximale dans le sac sans dépasser la capacité du sac-à-dos.

I.1. Proposer une modélisation de ce problème (Fixer des notations, des paramètres, des structures,...).

Une première approche consiste à mettre les objets dans le sac par "rentabilité" décroissante, la "rentabilité" étant le rapport entre la valeur et le poids.

I.2. Justifier empiriquement cette approche.

I.3. Montrer que cette approche, dans certains cas, ne fournit pas la meilleure solution.

Une solution "bulldozer"

I.4. Écrire un algorithme basé sur l'énumération qui donne toutes les solutions optimales (maximisant la valeur du sac).

I.5. Donner l'ordre de grandeur de sa complexité.

Une solution plus fine inspirée du principe de *programmation dynamique*

I.6. En notant $V(k, c)$ la valeur optimale du sac de capacité c obtenue avec les k premiers objets, écrire une équation de récurrence vérifiée par les $V(k, c)$.

I.7. Écrire un algorithme résolvant le problème.

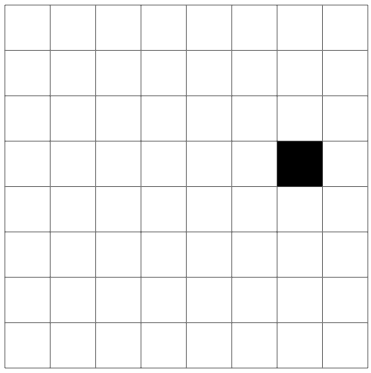
On considère l'instance suivante de 6 objets pour un sac de capacité 11 :

objet	poids	valeur
1	2	5
2	3	7
3	5	16
4	4	13
5	7	19
6	3	10

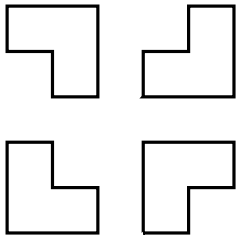
I.8. Expliquer votre algorithme en l'exécutant sur cet exemple.

II : Pavage

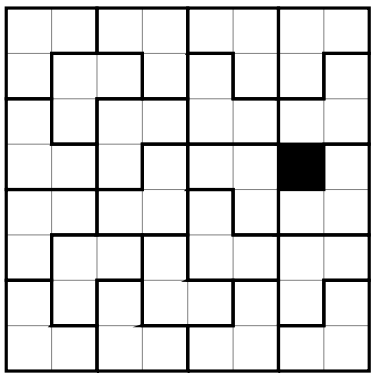
L'objectif est de réaliser le pavage d'un carré troué avec des pièces identiques en forme de L. Pour simplifier l'analyse, on suppose que le carré est de dimension $2^k \times 2^k$, le trou est dans une position arbitraire.



(a) Un exemple de carré à paver



(b) Formes



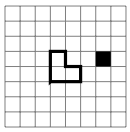
(c) Un exemple de pavage

II.1. Résoudre le problème de pavage pour $k = 1$ (carré 2×2).

II.2. Résoudre le problème pour $k = 2$ (carré 4×4)

II.3. Écrire un algorithme (récursif), reposant sur le principe du *diviser pour régner*, résolvant le problème de pavage d'un carré troué $2^k \times 2^k$.

Indication : Commencer par placer un L au centre du carré



II.4. Donner le pavage obtenu après l'exécution de l'algorithme sur l'exemple ci-dessus. On indiquera sur chaque pièce l'ordre dans lequel elle a été posée.

II.5. Calculer la complexité de cet algorithme.

II.6. Cette méthode peut-elle s'appliquer à un carré quelconque de dimension $n \times n$?