

**INF550. Conception et analyse d'algorithmes**

Examen du vendredi 5 décembre 2008. 2 heures

Documents autorisés : poly, transparents du cours, énoncés de PC

*Le barême est indicatif. On vous demande d'accorder une attention particulière à une rédaction soignée, qui sera un critère important dans l'évaluation.*

Vous êtes embauché par une multinationale de renommée internationale pour inclure dans la prochaine version de son système d'exploitation une puissante suite logicielle d'aide à la pose de carreaux de salle de bain pour clients excentriques.

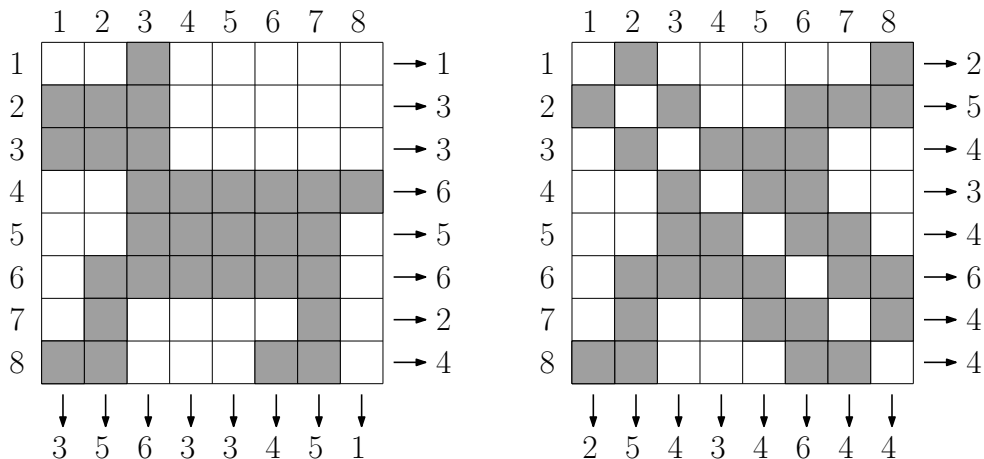


FIG. 1 – (a) un carrelage de taille  $8 \times 8$ . (b) un carrelage symétrique de taille  $8 \times 8$ .

### I. Algorithme de flot maximum

Un client désire que soient posés des carreaux unité ( $1 \times 1$ ) blancs et noirs sur un mur de taille  $n \times n$ . Il souhaite de plus choisir le nombre de carrés noirs dans chaque ligne et dans chaque colonne. Vous devez proposer un carrelage qui satisfait ces conditions.

Il s'agit donc, étant donnés deux vecteurs de  $n$  entiers  $\ell = (\ell_1, \dots, \ell_n)$  et  $c = (c_1, \dots, c_n)$ , de déterminer une matrice  $M = (m_{i,j})_{1 \leq i,j \leq n}$  dont les entrées sont dans  $\{0, 1\}$  et telle que  $\sum_{i=1}^n m_{i,j} = c_j$  pour tout  $j = 1, \dots, n$  et  $\sum_{j=1}^n m_{i,j} = \ell_i$  pour tout  $i = 1, \dots, n$ . On suppose que  $\ell_1 + \dots + \ell_n = c_1 + \dots + c_n$ , sans quoi il n'y a évidemment pas de solution.

1. Utiliser la figure 1(a) pour donner un exemple de carrelage ayant comme vecteurs de projection  $\ell = (1, 2, 3, 3, 4, 5, 6, 6)$  et  $c = (1, 3, 3, 3, 4, 5, 5, 6)$ .
2. Modéliser ce problème par un problème de flot maximum dans un graphe ayant  $2n + 2$  sommets (source et puits compris). Quelle est la complexité d'un algorithme qui propose un carrelage par cette méthode ?
3. Soient  $\ell^* = (\ell_1^*, \dots, \ell_n^*)$  et  $c^* = (c_1^*, \dots, c_n^*)$  les vecteurs obtenus en réordonnant les  $\ell_i$  et  $c_i$  en ordre croissant. Montrer, en utilisant le théorème FlotMax=CoupeMin qu'il existe un carrelage satisfaisant si et seulement si pour tout  $r$  et  $s$  avec  $1 \leq r, s \leq n$  on a

$$\sum_{j=s+1}^n c_j^* - \sum_{i=1}^r \ell_i^* \leq rs.$$

*Commentaire "culturel".* Dans le cas où l'on donne, en plus des vecteurs  $\ell$  et  $c$ , un vecteur  $\delta$  qui indique le nombre de carreaux noirs dans chaque diagonale, on ne connaît pas d'algorithme polynomial en  $n$  qui décide s'il existe une matrice ayant  $\ell$ ,  $c$  et  $\delta$  comme projection. On peut montrer, mais la réduction est assez complexe, que ce problème est NP-complet. Ce type de problèmes est en réalité motivé par des questions de reconstruction d'images à partir de projections (radiographies ou RMN en imagerie médicale, cristallographie, etc).

## II. Couplages, algorithme glouton et connexité

Le client exige de plus qu'il n'y ait pas de carré noir sur la première diagonale et que la figure formée par les carrés noirs et blancs soit symétrique par rapport à cette diagonale (en particulier il faut que  $c = \ell$ ). Un exemple est donné à la figure 1(b).

1. Montrer que le problème est équivalent à la recherche d'un graphe  $H$  non orienté à  $n$  sommets de degrés respectifs  $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$  (le degré d'un sommet est son nombre de voisins). Calculer le nombre  $m$  d'arêtes d'un tel graphe.
2. Modéliser ce problème par la recherche d'un couplage dans le graphe  $G = (X, E)$  avec  $X = X_1 \cup X_2$  et  $E = E_1 \cup E_2$  où
  - $X_1 = \{v_{i,j} \mid 1 \leq i \neq j \leq n\}$ , de sorte que  $|X_1| = n(n-1)$ ,
  - $X_2 = \{w_{i,k} \mid 1 \leq i \leq n, 1 \leq k \leq n - d_i\}$ , de sorte que  $|X_2| = \sum_i (n - d_i)$ ,
  - $E_1 = \{\{v_{i,j}, v_{j,i}\} \mid 1 \leq i \neq j \leq n\}$ , de sorte que  $|E_1| = n(n-1)$ ,
  - $E_2 = \{\{v_{i,j}, w_{i,k}\} \mid 1 \leq i \neq j \leq n, 1 \leq k \leq n - d_i\}$ , de sorte que  $|E_2| = \sum_i (n-1)(n - d_i)$ .

On cherche une caractérisation des suites d'entier qui sont suite de degrés d'un graphe.

2. Montrer que si les sommets d'un graphe ont pour degré  $d_1 \geq \dots \geq d_n \geq 0$  alors

$$\sum_{i=1}^k d_i \leq k(k-1) + \sum_{i=k+1}^n \min(k, d_i).$$

En fait cette condition est suffisante pour que les  $d_i$  soient la suite des degrés d'un graphe, mais la preuve étant assez longue on admettra ce résultat.

Le graphe solution trouvé par la réduction à un problème de couplage n'est pas forcément connexe. Quelques rappels : deux sommets  $x$  et  $y$  d'un graphe  $H$  sont *connectés* dans  $H$  s'il existe un chemin de  $x$  à  $y$  dans  $H$  ; une *composante connexe* de  $H$  est un ensemble maximal de sommets 2 à 2 connecté dans  $H$  ; un graphe est *connexe* s'il ne possède qu'une seule composante connexe ; un graphe connexe possède (au moins) un arbre couvrant.

3. À partir de la comparaison du nombre de sommets et du nombre d'arêtes d'un graphe connexe, donner une condition nécessaire sur la somme des  $d_i$  pour qu'il puisse exister un graphe solution connexe.
4. Étant donné un graphe non connexe satisfaisant la condition précédente, proposer une modification locale d'une paire d'arêtes bien choisies qui diminue le nombre de composantes connexes sans changer les degrés des sommets.

En déduire un algorithme pour transformer un graphe non connexe de suite de degrés  $d_1 \geq \dots \geq d_n$  en un graphe connexe de même suite de degrés. Quelle est la complexité de votre algorithme ?

5. Déduire des résultats précédents une condition nécessaire et suffisante sur une suite  $d_1 \geq \dots \geq d_n \geq 0$  pour qu'elle soit la suite des degrés d'un graphe connexe.

### III. Programmation dynamique

Un client désire un carrelage de taille  $m \times n$  formé de carrés unité blancs et d'une frise noire, constituée de carreaux rectangulaires de taille  $1 \times k$  pour différentes valeurs de  $k$ . La frise part du carré  $(1, 1)$  et arrive au carré  $(m, n)$  et doit progresser de gauche à droite lorsqu'on la regarde tournée de  $45^\circ$  (pas de retour en arrière). Elle doit de plus éviter des obstacles indiqués par une matrice booléenne  $P = (p_{i,j})_{1 \leq i,j \leq n} : p_{i,j}$  vaut 1 si on peut poser un carreau en position  $(i, j)$ , 0 sinon.

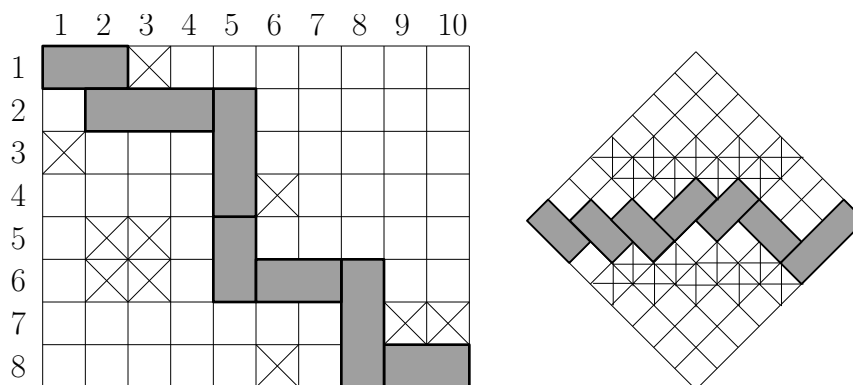


FIG. 2 – (a) une frise de taille  $8 \times 10$       (b) une frise  $8 \times 8$ , tournée de  $45^\circ$

1. On suppose dans un premier temps qu'on dispose d'une réserve infinie de chaque type de carreaux noirs : les carreaux de type  $i$  sont de taille  $1 \times \ell_i$  et de coût  $c_i$ , pour  $i = 1, \dots, k$ . Donner un algorithme polynomial qui propose une frise réalisable de coût minimum. Quel est la complexité de votre algorithme ?
2. On suppose maintenant qu'on ne dispose que d'une réserve finie de  $k$  carreaux : le  $i$ ème carreau est de forme  $1 \times \ell_i$  et à un coût  $c_i$ . On cherche parmi les frises réalisables avec les réserves disponibles une de coût minimum. Donner pour cela un algorithme polynomial en  $m, n$  et  $k$ .

## IV. NP-complétude

On reprend tout d'abord le problème de l'exercice précédent dans le cas simplifié où il n'y a pas d'obstacle. Autrement dit on considère le problème FRISE suivant :

**Donnée :** la taille  $(m, n)$  de la grille et une liste  $(\ell_1, c_1), \dots, (\ell_k, c_k)$  de  $k$  pièces (de forme  $1 \times \ell_i$  et de coût  $c_i$ ) ;

**Problème :** Trouver si elle existe une frise joignant  $(1, 1)$  à  $(m, n)$  de coût minimum.

On remarque alors que les données ( $n$ ,  $m$  et la liste des  $k$  pièces disponibles) sont de taille  $N = O(k(\log(n + m + C)))$  où  $C$  est le coût maximum d'une pièce : on se pose le problème de l'existence d'un algorithme polynomial en  $N$  pour résoudre FRISE.

1. Énoncez le problème de décision associé à la recherche d'une frise de coût minimum.
2. Montrez que ce problème est NP-complet.

Pour montrer la NP-complétude du problème mentionné à la fin de l'exercice I, une première étape est de montrer la NP-complétude du problème TRIPARTITION suivant :

**Donnée :** des entiers positifs  $x_1, x_2, \dots, x_n$

**Problème :** existe-t-il une partition de  $\{1, \dots, n\}$  en trois parties  $X_1, X_2, X_3$  telle que  $\sum_{j \in X_i} x_j = \frac{1}{3} \sum_{j=1}^n x_j$  pour  $i = 1, 2, 3$ .

3. Montrez que ce problème TRIPARTITION est NP-complet.