L3 Informatique **année 2012-2013**

Méthodes Algorithmiques

Examen du 26 Juin 2013
Responsable : Sophie Pinchinat

Seule une feuille manuscrite en format A4 recto-verso est autorisée.
Le barème est donné à titre indicatif.

Exercice 1 : Plus grand carré de 1 (14 points)

On considère le problème suivant :

Nom :	PLUS-GRAND-CARRÉ-DE-1
Entrée :	Une matrice A de taille $n \times m$ où les coefficients valent 0 ou 1.
Sortie :	La largeur maximum $K \geq 0$ d'un carré de 1 dans A, ainsi que pour le cas où $K \neq 0$ les coordonnées (I, J) du coin en haut à gauche d'un tel carré (autrement dit pour tous $i, j, I \leq i \leq I + K - 1$, $J \leq j \leq J + K - 1$, $A[i, j] = 1$ et le point (I, J) est le plus "à gauche puis en haut").

La Figure 1 montre un exemple d'une matrice pour laquelle la réponse au problème est $K = 2$. Or, il y a deux carrés de 1 dans A de taille 2, aux coordonnées $(2, 3)$, mais aussi $(1, 2)$ et $(2, 4)$. La réponse est toutefois $(1, 2)$ car c'est la case la plus "à gauche puis en haut".

A	1	2	3	4	5
1	0	1	1	0	1
2	0	1	1	1	1
3	0	0	1	1	1

Figure 1: Le carré le plus large est de taille 2.

On se propose de développer une solution basée sur la programmation dynamique.

1. En préliminaire de la résolution du problème, on formalise l'ordre "à gauche puis en haut" entre les cases de la matrice A. Soient $1 \leq i, i' \leq n$ et $1 \leq j, j' \leq m$, on note $(i, j) \preceq (i', j')$ pour signifier que la case (i, j) est plus "à gauche puis en haut" que la case (i', j').

   Donnez une expression analytique qui caractérise $(i, j) \preceq (i', j')$.

   Illustrez l'ordre $\preceq$ pour les trois cases $(2, 3)$, $(1, 2)$ et $(2, 4)$.

   Dans la suite de cet exercice, vous pourrez utiliser l'ordre $\preceq$ sans le redéfinir.

2. On note désormais $Carre[i,j]$ la réponse (I,J,K) au sous-problème posé pour la sous-matrice $A[i..n, j..m]$.

   Dans notre exemple, on a $Carre[3,5] = (3,5,1)$ et $Carre[1,4] = (2,4,2)$.

   Notations : Dans la suite, si $Carre[i,j] = (I', J', K')$, on écrira simplement $Carre[i,j].(I,J)$ (respectivement $Carre[i,j].K$) pour (I', J') (respectivement K'), de sorte que par exemple $Carre[3,5].(I,J) = (3,5)$ et $Carre[3,5].K = 1$.

   Expliquer ce que vaut $Carre[i,m]$ pour $1 \leq i \leq n$.

   Quelles autres valeurs de $Carre[i,j]$ pouvez-vous aussi facilement calculer ?

3. Soit un couple (i,j) fixé tel que $A[i,j] = 0$.

   Exprimez $Carre[i,j].K$ en fonction de $Carre[i+1,j].K$, $Carre[i,j+1].K$ et $Carre[i+1,j+1].K$.

4. Soit encore un couple (i,j) fixé tel que $A[i,j] = 0$.

   En considérant l'ordre $\preceq$ entre les cases $(i+1,j)$, $(i,j+1)$ et $(i+1,j+1)$, expliquez comment vous déduisez la valeur $Carre[i,j].(I,J)$ de la façon dont $Carre[i,j].K$ a été obtenue.

5. On considère maintenant un couple (i,j) tel que $A[i,j] = 1$.

   Comment se calcule $Carre[i,j].K$?

   Indication : on distinguera selon que

   $$A[i',j'] = 1, \text{ pour tout } (i',j') \in \{(i+1,j), (i,j+1), (i+1,j+1)\}$$

6. Toujours dans le cas d'un couple (i,j) tel que $A[i,j] = 1$, expliquez comment se calcule la valeur $Carre[i,j].(I,J)$.

7. Déduisez des questions qui précèdent un algorithme basé sur la programmation dynamique qui répond au problème Plus-grand-carré-de-1.

   Indications : on pourra regrouper le cas $A[i,j] = 0$ avec celui où l'un parmi $A[i+1,j]$, $A[i,j+1]$ et $A[i+1,j+1]$ a la valeur 0.

8. Exécutez votre algorithme sur l'exemple de la Figure 1.

9. Quelle est la complexité de votre algorithme ?

Exercice 2 : Le problème du gymnase (6 points)
On considère un gymnase dans lequel se déroulent de nombreuses épreuves : on souhaite en “planifier” le plus possible, sachant que deux événements ne peuvent avoir lieu en même temps (puisqu'il n'y a qu'un gymnase). Un événement i est caractérisé par une date de début d_i et une date de fin f_i. On dit que deux événements sont compatibles si leurs intervalles de temps sont disjoints. On veut résoudre le problème à l'aide d'un programme glouton.

1. Essai 1 : On trie les événements par durée croissante puis on “gloutonne”, c-à-d. qu'on met les plus courts en premier s'ils sont compatibles avec ceux déjà placés.

2. Essai 2 : Plutôt que de trier les événements par durée croissante, on peut les classer par date de commencement croissante.

3. Essai 3 : On trie les événements par ceux qui intersectent le moins possible d'autres événements.

4. Essai 4 : On peut enfin trier les événements par date de fin croissante.

Pour chacun des différents essais, répondez si l'algorithme est optimal. Si votre réponse est “oui” démontrez-le. Dans le cas contraire, justifiez par un (contre) exemple.

Indications : Pour le cas de l'Essai 4, vous pourrez vous inspirer de l'exemple CHOIX-D'ACTIVITÉS *étudié en cours.*