Algorithmique

Partiel du 17 novembre 2014

durée 2 heures 30

Le polycopié du cours est le seul document autorisé.

Les exercices sont indépendants.

Les procédures et fonctions devront être écrites en pseudo-code (pas en Java, Caml, ...)

Toutes les réponses devront être correctement justifiées. La rigueur des raisonnements, la clarté des explications, et la qualité de la présentation influeront sensiblement sur la note.

Le chiffre en regard d'une question est une indication sur sa difficulté ou sa longueur.

1 Preuve de Hoare

On considère le programme suivant ($t[1 \cdot\cdot n]$ est un tableau d'entiers) :

```
j ← n
tant que j > 0 faire
   i ← 1
   tant que i < j faire
      si t[i] > t[i+1] alors Echanger(t[i], t[i+1]) fsi
      i ← i + 1
   ftq
   j ← j − 1
ftq
```

[4] **a)** Montrer que la boucle interne (sur i) satisfait l'invariant : $t[1 \cdot\cdot i-1] \leq t[i]$.
Prouver avec la méthode de Hoare que ce programme réalise un tri du tableau t.
Donner bien tous les invariants utilisés pour votre preuve.

2 Complexité

On considère le programme suivant ($t[1 \cdot\cdot n]$ est un tableau d'entiers) :

```
pour i ← 1 à n − 1 faire
   si t[i] > t[i+1] alors Echanger(t[i], t[i+1]) fsi
fpour
```

Soit $X : \mathfrak{S}_n \to \mathbb{N}$ la variable aléatoire telle que $X(\sigma)$ est le nombre de fois où l'instruction `Echanger`$(-,-)$ est exécutée par le programme ci-dessus lorsque le tableau t est initialisé avec σ. L'objectif est de calculer $E(X)$.

[1] **a)** Soit $1 \leq k \leq n$. Montrer que $|A_k| = \frac{n!}{k}$ où $A_k = \{\sigma \in \mathfrak{S}_n \mid \sigma(k) = \max(\sigma(1), \dots, \sigma(k))\}$.
La preuve est simple, ne vous lancez pas dans des calculs compliqués.

[3] **b)** Montrer que $E(X) = n - \sum_{k=1}^{n} \frac{1}{k}$.
Indication : pour tout $k \in \{1, \dots, n-1\}$, calculer l'espérance de la variable aléatoire $X_k : \mathfrak{S}_n \to \{0,1\}$ définie par $X_k(\sigma) = 1$ si et seulement si l'instruction `Echanger`$(t[k], t[k+1])$ est exécutée par le programme ci-dessus lorsque le tableau t est initialisé avec σ.

3 Relais pour téléphones mobiles

On considère une route *droite* sur laquelle se trouvent n maisons. On cherche à placer des relais pour téléphones mobiles le long de la route de telle sorte que chaque maison se trouve à une distance au plus K d'un relais. On veut bien sûr minimiser le nombre de relais utilisés.

Formellement, une donnée du problème est une suite de réels $x_1 < x_2 < \cdots < x_n$. Une solution est une suite $y_1, y_2, \ldots, y_m$ de réels telle que

$$\forall\, 1 \leq i \leq n,\ \exists\, 1 \leq j \leq m, \qquad |x_i - y_j| \leq K. \tag{1}$$

Une solution est optimale si elle utilise un nombre minimal de relais (m minimal).

[4] **a)** Proposer un algorithme pour trouver une solution optimale en temps $\mathcal{O}(n)$.
Montrer que la solution fournie par votre algorithme est correcte, i.e., vérifie (1).
Montrer que la solution fournie par votre algorithme est optimale.

4 Multiplication de matrices booléennes

Dans ce problème, la numérotation des lignes et colonnes d'une matrice commence à 0. Si B est une matrice $k \times n$, on note $B[i,j]$ le coefficient i, j de la matrice B (avec $0 \leq i < k$ et $0 \leq\ j < n$). Une matrice est booléenne si ses coefficients sont 0 ou 1.

Soit $k > 0$ un entier positif. On note $S^{(k)}$ la matrice booléenne $2^k \times k$ telle que pour tout $a \in \{0, \ldots, 2^k - 1\}$, la ligne a de $S^{(k)}$ est l'écriture binaire de a en commençant par le bit de poids faible : si $S^{(k)}[a,j] = a_j$ pour $0 \leq j < k$ alors $a = \overline{a_{k-1} \cdots a_1 a_0}^2$. Par exemple,

$$A^{(2)} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

[4] **a)** Soit B une matrice $k \times n$ arbitraire.
Pour $k > 1$, donner une décomposition en blocs de $S^{(k)}$ qui fasse intervenir $S^{(k-1)}$.
Exprimer le produit $C = S^{(k)} \times B$ par blocs.
Montrer que l'on peut calculer le produit $C = S^{(k)} \times B$ en temps $\mathcal{O}(2^k n)$.
Écrire l'algorithme *itératif* qui réalise ce calcul.

[2] **b)** Soit A une matrice $m \times k$ *booléenne* et soit B une matrice $k \times n$ arbitraire. Montrer que l'on peut calculer la matrice $A \times B$ en temps $\mathcal{O}(mk + mn + 2^k n)$.

[1] **c)** Soit A une matrice $m \times (\ell k)$ *booléenne* et soit B une matrice $(\ell k) \times n$ arbitraire. Montrer que l'on peut calculer la matrice $A \times B$ en temps $\mathcal{O}(\ell(mk + mn + 2^k n))$.

[1] **d)** Soit A une matrice $n \times n$ *booléenne* et soit B une matrice $n \times n$ arbitraire. Montrer que l'on peut calculer la matrice $A \times B$ en temps $\mathcal{O}(\frac{n^3}{\log_2 n})$.