Algorithmique
Examen Partiel
26 octobre 2017 13h30–15h30

Documentation autorisée : deux feuilles A4 recto verso manuscrites. Tout autre document ou appareil est interdit. **Les téléphones portables doivent être éteints et rangés dans votre sac, et vous devez poser sacs et vestes à l'avant de la salle avant de commencer.**

> **Indications :** le barème est donné à titre indicatif seulement. Sauf indication contraire, justifiez vos réponses et expliquez vos algorithmes. Les algorithmes doivent être aussi efficaces que possible et présentés de façon lisible et claire. Cela veut dire que les noms de variables doivent être bien choisis pour représenter ce qu'elles sont censées contenir et que les principales étapes doivent être accompagnées d'explications. Une partie de la note portera sur la clarté de présentation de vos réponses.

Exercice 1 : (6 points)

Une clique dans un graphe est un sous-ensemble de sommets tous reliés les uns aux autres par des arêtes du graphe.

1. Donner une récurrence structurelle qui exprime la *taille* d'une clique maximale d'un graphe en fonction de la taille de cliques maximales dans des graphes plus petits.
2. Donner un algorithme de type *backtracking* qui prend en entrée un graphe G et retourne la taille d'une clique de taille maximale de G. (Il n'est pas nécessaire de retourner la clique elle-même). Votre algorithme doit implémenter la récurrence donnée précédemment.
3. Donner une récurrence pour la complexité de votre algorithme.

Noter qu'il n'est pas nécessaire pour avoir tous les points de donner un algorithme dont la complexité est très optimisée, ni de résoudre la récurrence.

Exercice 2 : (6 points)

On propose un algorithme pour multiplier deux entiers longs, qui applique "l'astuce de Karatsuba" deux fois de la façon suivante : posons $\ell = \lceil \frac{N}{3} \rceil$ où x et y sont des entiers de taille N représentés en binaire. Décomposons x et y en 3 parties telles que

$$x = 2^{2\ell}x_2 + 2^{\ell}x_1 + x_0 \quad \text{et} \quad y = 2^{2\ell}y_2 + 2^{\ell}y_1 + y_0$$

avec pour chaque partie $0 \leqslant x_i < 2^{\ell}$ et $0 \leqslant y_i < 2^{\ell}$. En développant, le produit s'écrit

$$xy = 2^{4\ell}(x_2y_2) + 2^{3\ell}\underline{(x_1y_2 + x_2y_1)} + 2^{2\ell}(x_2y_0 + x_1y_1 + x_0y_2) + 2^{\ell}\underline{(x_1y_0 + x_0y_1)} + x_0y_0.$$

En posant

$$M_{12} = (x_2 - x_1)(y_1 - y_2) \quad \text{et} \quad M_{01} = (x_1 - x_0)(y_0 - y_1),$$

on observe que

$$\underline{(x_1y_2 + x_2y_1)} = M_{12} - x_2y_2 - x_1y_1 \quad \text{et} \quad \underline{(x_1y_0 + x_0y_1)} = M_{01} - x_1y_1 - x_0y_0,$$

et on obtient ainsi la récurrence suivante :

$$xy = 2^{4\ell}(x_2y_2)+2^{3\ell}(M_{12}-x_2y_2-x_1y_1)+2^{2\ell}(x_2y_0+x_1y_1+x_0y_2)+2^{\ell}(M_{01}-x_1y_1-x_0y_0)+x_0y_0$$

Appelons *Karatsuba3* l'algorithme diviser-pour-régner qui implémente cette récurrence. Nous nous intéressons à sa complexité en fonction de la taille des entiers considérés, c'est-à-dire la taille de leur représentation en binaire ; les opérations sur les entiers étant faites en binaire, la division et la multiplication par une puissance de 2 sont de simples décalages qui s'effectuent en temps linéaire, de même que l'addition.

1. Rappeler la récurrence de complexité de l'algorithme de Karatsuba ("avec l'astuce") vu en cours, ainsi que sa complexité.
2. Donner, pour un appel de l'algorithme Karatsuba3 sur des entiers de taille N (sans compter les opérations effectuées pendant les appels récurdifs), les nombres
   (a) d'additions, (b) de décalages, et (c) d'appels récursifs.
   Préciser la taille des entiers multipliés récursivement.
3. Donner une récurrence pour la complexité de ce nouvel algorithme, en fonction de la taille des entrées (vous pouvez supposer $|x| = |y| = N$).
4. Résoudre la récurrence pour donner la complexité de l'algorithme Karatsuba3. Justifier votre réponse.
5. Quel algorithme, Karatsuba, ou Karatsuba3, a la meilleure complexité ? Justifier votre réponse.

Exercice 3 : (4 points)

Résoudre les récurrences suivantes. (Donner uniquement la réponse.)

1. $T(N) = 7T(N/8) + N$
2. $T(N) = 7T(N/8) + N^3$
3. $T(N) = 4T(N/4) + \sqrt{N}$
4. $T(N) = 4T(N/2) + N^2$
5. $T(N) = T(N-2) + T(N-4)$

Exercice 4 : (4 points)

Nous avons vu qu'un problème de décision pour un langage L est dans la classe $\mathcal{NP}$ s'il existe un prédicat de vérification $\mathtt{V}$ décidable en temps polynomial tel que $X \in L$ si et seulement s'il existe un certificat π tel que $\mathtt{V}(X, \pi)$ est vrai.
Décrire un prédicat pour chacun des deux problèmes suivants, et justifier qu'il est vérifiable en temps polynomial en décrivant un algorithme adapté dont vous donnerez la complexité en fonction du nombre de sommets N et/ou du nombre d'arêtes M. Vous pouvez supposer que le graphe est donné sous forme de matrice d'adjacence $N \times N$.

1. CLIQUE
2. HAMILTONIANCYCLE

Annexe

Identités sur les logarithmes

- $\log_u(v) = \log_2(v)/\log_2(u)$
- $\log_a(b) < \log_a(c) \iff b < c$
- $\log_a(b) < \log_c(b) \iff a > c$

Master Theorem

Théorème. *Soit T une fonction vérifiant une relation de récurrence de la forme*

$$T(n) = a\,T(\tfrac{n}{b}) + f(n).$$

Alors (version du cours) :

$$\begin{array}{rl}
(\textbf{cas « } f \textbf{ petit »}) & af(n/b) = cf(n),\ c > 1 \implies T(n) \in \Theta(n^{\log_b a}) \\
(\textbf{cas « } f \textbf{ moyen »}) & af(n/b) = f(n) \implies T(n) \in \Theta(n^{\log_b a}\log n) \\
(\textbf{cas « } f \textbf{ grand »}) & af(n/b) = cf(n),\ c < 1 \implies T(n) \in \Theta(f(n))
\end{array}$$

Ou (version du TD) :

$$\begin{array}{rl}
(\textbf{cas « } f \textbf{ petit »}) & f(n) \in O(n^{\log_b a - \varepsilon}) \implies T(n) \in \Theta(n^{\log_b a}) \\
(\textbf{cas « } f \textbf{ moyen »}) & f(n) \in \Theta(n^{\log_b a}) \implies T(n) \in \Theta(n^{\log_b a}\log n) \\
(\textbf{cas « } f \textbf{ grand »}) & \left.\begin{array}{l} f(n) \in \Omega(n^{\log_b a + \varepsilon}) \\ af(\frac{n}{b}) \leqslant cf(n)\ \ (c < 1) \end{array}\right\} \implies T(n) \in \Theta(f(n))
\end{array}$$

Problèmes vus en cours et en TD

CIRCUIT-SAT : étant donné un circuit booléen C, existe-t-il une valuation de ses variables d'entrée telle que la sortie du circuit s'évalue à VRAI ?

3COL : étant donné un graphe G, ses sommets peuvent-ils être coloriés avec 3 couleurs de sorte que toutes les arêtes ait deux couleurs différentes à leurs extrémités ?

2COL : étant donné un graphe G, ses sommets peuvent-ils être coloriés avec 2 couleurs de sorte que toutes les arêtes ait deux couleurs différentes à leurs extrémités ?

INDEPENDENTSET : étant donné un graphe G et un entier k, G possède-t-il un ensemble *indépendant* de sommets de cardinal k (tel qu'il n'existe pas d'arêtes entre eux) ?

CLIQUE : étant donné un graphe G et un entier k, G possède-t-il une k-*clique* (un sous-graphe isomorphe au graphe complet à k sommets) ?

VERTEXCOVER : étant donné un graphe G et un entier k, G possède-t-il un ensemble *couvrant* C de k sommets (tel que toute arête soit incidente à au moins un des sommets de C) ?

SETCOVER : étant donné une collection d'ensembles $\mathcal{S} = \{S_1, \ldots, S_n\}$ et un entier k, existe-t-il une sous-collection de $\mathcal{S}$ de cardinal k dont l'union des éléments couvre celle des éléments de $\mathcal{S}$?

HAMILTONIANCYCLE : étant donné un graphe G, existe-t-il un cycle *hamiltonien* dans G (visitant chaque sommet exactement une fois) ?