IFT 2121 H00 **Pierre McKenzie**
Introduction à l'algorithmique

EXAMEN FINAL

Le samedi 15 avril 2000 de 9:00 à 12:00, à la patinoire

Directives:

- Aucune documentation. *No documentation.*
- Répondez sur le questionnaire. *Answer on the question sheets.*
- Lisez rapidement toutes les questions avant de commencer.
  Take a quick glance at all the questions before you start.
- La traduction anglaise de quelques questions est donnée en dernière page de cet examen. Cette dernière page ne sera pas considérée dans la correction.
  The English translation of a few selected questions appears on the last page of the exam. Do not write your answers on that last page, which will be discarded.

1. _______ /20
2. _______ /15
3. _______ /15
4. _______ /10
5. _______ /22
6. _______ /18

Total: _______ /100

Nom: ____________________________________ **Code permanent:** _________________

1. **(20 points)** Nous considérons des fonctions $f, g, t : \mathbb{N} \to \mathbb{R}^{\geq 0}$.

   (a) (4 pts) Donnez deux fonctions f et g telles que $\Omega(f)$ est un sous-ensemble strict de $\Omega(g)$.

   $f(n) =$ $\qquad\qquad\qquad\qquad$ $g(n) =$

   (b) (6 pts) Donnez sans justification une fonction explicite $f(n)$ telle que $t(n) \in \Theta(f(n))$, si

   i. $t(n) \in 9t(\lceil n/3 \rceil) + \Theta(n^2)$ $\qquad\qquad$ $f(n) =$

   ii. $t(n) \in 9t(\lceil n/3 \rceil) + \Theta(n)$ $\qquad\qquad$ $f(n) =$

   iii. $t(n) \in 9t(\lceil n/3 \rceil) + \Theta(\log n)$ $\qquad\qquad$ $f(n) =$

   (c) (5 pts) Vrai ou faux: Si $(f(n) + g(n))^2 \in O(n^3)$ alors $f(n) \in O(n^3)$ et $g(n) \in O(n^3)$. (Prouvez, ou donnez un contre-exemple.)

   (d) (5 pts) Vrai ou faux: Si $(f(n) + g(n))^2 \in \Theta(n^3)$ alors $f(n) \in \Theta(n^3)$ et $g(n) \in \Theta(n^3)$. (Prouvez, ou donnez un contre-exemple.)

2. **(15 points)** Soit G un graphe orienté de n sommets, donné par la matrice L telle que pour $1 \leq i, j \leq n$, on a $L[i, j] = vrai$ si l'arc (i, j) est présent, $L[i, j] = faux$ sinon. Donnez les détails d'un algorithme fonctionnant en temps $O(n^3)$ et calculant un tableau N où, pour $1 \leq i \leq n$, $N[i]$ est le *nombre de sommets* de G qui sont accessibles par un chemin à partir de i.
Indice. Un des algorithmes possibles utilise la programmation dynamique pour calculer d'abord une matrice $CHEMIN$ telle que, pour tout $1 \leq i, j \leq n$, $CHEMIN[i, j] = vrai$ si un chemin existe de i à j dans G, $CHEMIN[i, j] = faux$ sinon.

3. **(15 points)** Soient 5 matrices M_1, M_2, M_3, M_4, M_5, de dimensions respectives 2×4, 4×3, 3×5, 5×7, 7×2, dont on veut calculer le produit $M_1 M_2 M_3 M_4 M_5$.

   (a) (2 pts) Quelle est la dimension de $M_1 M_2 M_3 M_4 M_5$?

   (b) (10 pts) Donnez, avec justification (en donnant le tableau que produirait l'algorithme vu en cours pour résoudre ce problème), un ordre optimal dans lequel doit s'effectuer ce produit pour minimiser le nombre total de multiplications scalaires.

   (c) (3 pts) Quel est le nombre minimal de multiplications scalaires?

4. **(10 points)** NB: en présence d'un choix, visitez toujours le sommet de plus petite étiquette.

(a) (3 pts) Quels sont les trois types d'arcs hors de l'arborescence obtenue de la fouille en profondeur d'un graphe orienté?

(b) (7 pts) Tracez les arcs empruntés par une fouille en profondeur du graphe orienté ci-dessous, et identifiez chaque arc laissé hors de l'arborescence obtenue selon les 3 types que vous avez décrits ci-dessus.

5. **(22 points)** L'algorithme ci-dessous vise à déterminer si l'entier $m \geq 2$ est une somme de carrés, c'est à dire s'il existe des entiers $d_1, d_2, \ldots, d_k \geq 2$ tels que $m = \Sigma_{i=1}^{k} d_i^2$:

fonction *CARRÉS*(m : **naturel**) : **booléen**
  $d \leftarrow$ *uniforme*(2..$m/2$)
  si $d * d < m$ **alors retourner** *CARRÉS*($m - (d * d)$)
  sinon si $d * d = m$ **retourner** *vrai*
  sinon retourner *faux*

où *uniforme*(2..$m/2$) retourne un entier choisi avec probabilité uniforme parmi $\{2, \ldots, \lfloor m/2 \rfloor\}$.

(a) (3 pts) Cet algorithme est-il de Monte Carlo ou de Las Vegas? Justifiez.

(b) (6 pts) Dites si cet algorithme est vrai-biaisé, faux-biaisé, ou non biaisé. Expliquez.

(c) (2 pts) Sur l'exemplaire initial m, quelle est la suite de valeurs retournées successivement par *uniforme* qui prolongera au maximum l'exécution de *CARRÉS*(m)?

(d) (3 pts) Posez une récurrence décrivant le temps $T(m)$ de l'exécution de *CARRÉS*(m), en supposant la suite de choix identifiée en (c), et en supposant un coût unitaire pour chaque appel à *uniforme* et pour chaque opération arithmétique.

(e) (2 pts) Quelle fonction explicite simple $f(m)$ vérifie $T(m) \in \Theta(f(m))$?

(f) (6 pts) L'algorithme ci-dessous vise à déterminer si $m \geq 2$ *est* le carré d'un entier ≥ 2:

fonction *ESTCARRÉ*(m : **naturel**) : **booléen**
  $d \leftarrow$ *uniforme*$(2..m/2)$
  si $d * d = m$ **retourner** *vrai*
  sinon retourner *faux*.

L'algorithme *ESTCARRÉ* est-il p-correct pour une certaine constante p, $0 < p < 1$? Justifiez votre réponse.

6. **(18 points)** (15 pts) Donnez un algorithme, fonctionnant en un temps dans $O(\log n)$ sur une PRAM "EREW", qui calcule la somme des éléments aux positions $1, 3, 5, \ldots, 2\lceil n/2 \rceil - 1$ d'une liste de n entiers. La somme doit être placée dans une variable globale *SOMME* accessible à tous les processeurs.
   Comme d'habitude, on se permet d'associer un processeur à chaque élément de la liste et on se permet de détruire la liste. NB: merci d'être concis.

   (3 pts) Votre algorithme parallèle est-il optimal? (Justifiez.)

BONNE CHANCE ET BON ÉTÉ!

1a) Give f and g such that $\Omega(f)$ is a proper subset of $\Omega(g)$.
1b) Give without proof a function f as requested.
1c) "Si" means "If" and "alors" means "then". Prove or give a counter-example.
1d) "et" means "and".
2) G is a directed graph with n nodes given by the matrix L such that for $1 \leq i, j \leq n$, $L[i,j] = vrai$ if the arc (i,j) is present, $L[i,j] = faux$ otherwise. Give the details of an algorithm operating in $O(n^3)$ time and computing an array N of integers such that, for $1 \leq i \leq n$, $N[i]$ is the number of nodes *in G accessible by some path from node i.*
Hint. *One of the possible algorithms uses dynamic programming to first compute a matrix $CHEMIN$ such that, for all $1 \leq i, j \leq n$, $CHEMIN[i,j] = vrai$ if a path exists from i to j in G, $CHEMIN[i,j] = faux$ otherwise.*
3) The five matrices have the stated dimensions, respectively, and we wish to compute their product.
3a) What is the dimension of the product?
3b) Give, with justification (by giving the table produced by the algorithm seen in class), an optimal multiplication order for computing the product while minimizing the number of scalar multiplications.
3c) What the optimal number of scalar products?
4) When a choice arises, always traverse the node with smallest label.
4a) Which three types of arcs can arise outside of the forest obtained by a depth-first traversal of a directed graph?
4b) Draw the arcs encountered in a depth-first traversal of the graph, and identify the arcs left out of the traversal, according to the 3 types of arcs described in part (4a).
5) The given algorithm attempts to determine whether the integer $m \geq 2$ is a sum of squares, ie whether there exist integers $d_1, d_2, \ldots, d_k \geq 2$ such that $m = \Sigma_{i=1}^{k} d_i^2$, where uniforme$(2..m/2)$ *returns an integer uniformly randomly selected from $\{2, \ldots, \lfloor m/2 \rfloor\}$.*
5a) Is this a Monte Carlo or a Las Vegas algorithm? (Justify.)
5b) Is the algorithm true-biased, false-biased, or unbiased? (Justify.)
5c) On an initial instance m, which sequence of values returned by uniforme *will cause* CARRÉS(m) *to run for the longest time?*
5d) Give a recurrence describing the time $T(m)$ to execute CARRÉS(m), *assuming the sequence of choices which you identified in (5c), and assuming unit cost for each call to* uniforme *and for each arithmetic operation.*
5e) Which simple explicit function $f(m)$ satisfies $T(m) \in \Theta(f(m))$?
5f) The algorithm ESTCARRÉ *attempts to determine whether $m \geq 2$* is *the square of an integer ≥ 2. Is this algorithm p-correct for some constant p, $0 < p < 1$? (Justify.)*
6a) Give a EREW PRAM $O(\log n)$-time algorithm to compute the sum of the elements at positions $1, 3, 5, \ldots, 2\lceil n/2 \rceil - 1$ in a list of n integers. The sum must be placed in a variable SOMME *(accessible to each processor). As usual, we allow ourselves to associate a processor with each element in the list, and to destroy the list if we wish. NB: thanks for being brief.*
6b) Is your algorithm optimal? (Justify.)

GOOD LUCK!