IFT 2125 A2014
Introduction à l'algorithmique

Pierre McKenzie

EXAMEN FINAL

Le mercredi 10 décembre 2014 de 15:30 à 18:15

Directives :
— Aucune documentation. *No documentation.*
— Aucune calculatrice. *No calculator.*
— Répondez sur le questionnaire. *Answer on the question sheets, in the space provided within the French section of the exam. See the last two pages for an English translation. Do not write your answers on those two pages, which we will nonetheless recuperate at the end.*

1. ______ /30
2. ______ /18
3. ______ /18
4. ______ /17
5. ______ /17

Total : ______ /100

Nom : ________________________________ **Matricule :** ______________

Cochez si doctorant. ☐ *Tick if PhD student.*

1. **(30 points)** Nous considérons des fonctions $f, t : \mathbb{N} \to \mathbb{R}^{\geq 0}$.

   (a) (3 pts) Donnez la définition formelle complète de $O(f(n))$.

   (b) (3 pts) Vrai ou faux (justifiez) : lorsque $t \in \Omega(f)$, $O(t) \subseteq \Omega(f)$.

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

   i. $t(n) \in t(\lfloor n/2 \rfloor) + \Theta(\log n)$ $\qquad f(n) =$

   ii. $t(n) \in 2t(\lfloor n/2 \rfloor) + \Theta(\log n)$ $\qquad f(n) =$

   iii. $t(n) \in t(n-2) + \Theta(\log n)$ $\qquad f(n) =$

   (d) (3 pts) Vrai ou faux (justifiez). Toutes les façons de former le produit de matrices $ABCD$, où les dimensions de A, B, C, D sont 2×3, 3×4, 4×3 et 3×2 respectivement, nécessitent le même nombre de produits scalaires.

   (e) (3 pts) Vrai ou faux (justifiez). Un point d'articulation d'un graphe non orienté est défini comme un sommet auquel la fouille en profondeur rebrousse chemin.

(f) (3 pts) Vrai ou faux (justifiez) : il existe un algorithme de Monte Carlo $\frac{1}{2}$-correct décidant en temps constant si un entier m est premier.

(g) (3 pts) Quelle est la probabilité d'obtenir 2 succès en 3 tentatives si à chaque tentative la probabilité de succès est de $\frac{3}{5}$?

(h) (3 pts) Vrai ou faux (justifiez) : il existe un algorithme de Monte Carlo $\frac{1}{m}$-correct et vrai-biaisé décidant en temps polynomial si un entier $m > 0$ est composé.

(i) (3 pts) Vrai ou faux (justifiez). Il est possible de calculer le OU de n bits en temps constant à l'aide de n processeurs reliés à une mémoire centrale fonctionnant en mode ERCW ("exclusive read concurrent write").

2. **(18 points)** Considérez le problème suivant :

   <u>PAIR</u>
   DONNÉE: entier $m \geq 0$
   DÉCIDER: si m est pair, i.e, divisible par 2 sans reste.

   Vous disposez d'un langage de programmation dans lequel les seules opérations arithmétiques sont l'ajout de 1 et le retrait de 1 (donc pas de $+, -, \times, //, \%$ ni de décalages d'un nombre de bits). De même, une boucle *for* n'incrémente son index que de 1 à chaque tour de boucle. Vous pouvez toutefois initialiser une variable à une constante, tester l'égalité de deux variables, etc.

   (a) (6 pts) Donnez dans ce langage un algorithme $A(m)$ résolvant PAIR.

   (b) (3 pts) Estimez l'ordre exact du nombre $t_A(n)$ d'instructions exécutées par votre algorithme $A(m)$ en fonction du nombre n de bits de m.

   (c) (6 pts) Donnez maintenant dans le même langage un algorithme diviser-pour-régner $B(m)$ résolvant PAIR. (Inutile de chercher trop compliqué.)

   (d) (3 pts) Posez précisément la récurrence qui décrit le nombre $t_B(m)$ d'instructions exécutées par votre algorithme B en fonction de m.

3. **(18 points)** Rappelons le problème du SAC À DOS,

   Donnée : capacité $W \in \mathbb{R}^{\geq 0}$ et objets $1, 2, \ldots, n$ de poids $w_1, \ldots, w_n \in \mathbb{R}^{\geq 0}$ et de valeurs $v_1, \ldots, v_n \in \mathbb{R}^{\geq 0}$

   Calculer : sous-ensemble d'objets de valeur maximale et de poids n'excédant pas W (un objet ne peut être choisi plus d'une fois),

dont voici un exemplaire :

$W = 10$	objet 1	objet 2	objet 3	objet 4	objet 5
poids	1	2	4	4	4
valeurs	3	1	4	5	2

(a) (2 pts) Ce problème est-il résoluble en général par la technique vorace ? OUI NON

(b) (6 pts) Solutionnez l'exemplaire ci-dessus par l'approche vorace, en indiquant vos étapes. (Si vous avez répondu NON ci-dessus, *décrivez clairement avant de commencer* une variante du problème SAC À DOS que votre technique vorace solutionnera à coup sûr.)

Quelle est la valeur maximale trouvée ? $\boxed{}$

Même exemplaire répété :

$W = 10$	objet 1	objet 2	objet 3	objet 4	objet 5
poids	1	2	4	4	4
valeurs	3	1	4	5	2

(c) (3 pts) Quelle identité est au coeur de la stratégie de programmation dynamique appliquée à SAC À DOS ?

(d) (7 pts) Résolvez l'exemplaire ci-dessus à l'aide de la programmation dynamique, en indiquant l'ordre et les étapes de remplissage d'un tableau approprié.

Quelle est la valeur maximale trouvée ? $\boxed{}$

Quels éléments forment l'ensemble choisi ?

4. **(17 points)** Soit G un graphe connexe orienté, avec longueurs positives aux arcs. Pour tout sommet v de G, on définit l'excentricité de v (notée $E(v)$) comme :

$$E(v) = \max_{w \in G}\{\ l(w,v) \mid l(w,v) \text{ est la longueur minimum des chemins de } w \text{ à } v\}$$

Un sommet d'excentricité minimum est appelé un centre de G.

(a) (14 pts) Donnez un algorithme détaillé (si vous utilisez un algorithme vu en cours, alors il faut donner cet algorithme au long), faisant intervenir la programmation dynamique, calculant un centre de G.

(b) (3 pts) Estimez sans justification la complexité en pire cas de votre algorithme en fonction du nombre de sommets :

5. **(17 points)** Le problème du coloriage d'un graphe est le suivant :

<u>COLORIAGE</u>

DONNÉE: matrice d'adjacence M d'un graphe non orienté de m sommets

DÉCIDER: s'il existe un vecteur $v \in \{\text{bleu, rouge, vert}\}^m$ tel que pour toute arête (a, b) du graphe, $v[a] \neq v[b]$.

(a) (14 pts) Esquissez un algorithme à retour arrière colo(M) pour COLORIAGE.
Indice. Pensez à ce que serait un vecteur k-prometteur, $0 \leq k \leq m$.

(b) (3 pts) Même si COLORIAGE ne possède pas d'algorithme déterministe polynomial connu, pourriez-vous transformer votre algo ci-dessus en un algorithme de Las Vegas fonctionnant en temps polynomial ? (Expliquez.)

English translations. Please do not write your answers here.

1. **(30 points)** We consider functions $f, t : \mathbb{N} \to \mathbb{R}^{\geq 0}$.
   (a) (3 pts) Give the complete formal definition of $O(f(n))$.
   (b) (3 pts) True or false (justify) : when $t \in \Omega(f)$, $O(t) \subseteq \Omega(f)$.
   (c) (6 pts) Give without justification a function $f(n)$ such that $t(n) \in \Theta(f(n))$, when
      i. $t(n) \in t(\lfloor n/2 \rfloor) + \Theta(\log n)$ $\qquad f(n) =$
      ii. $t(n) \in 2t(\lfloor n/2 \rfloor) + \Theta(\log n)$ $\qquad f(n) =$
      iii. $t(n) \in t(n-2) + \Theta(\log n)$ $\qquad f(n) =$
   (d) (3 pts) True or false (justify). Every bracketing of the matrix product $ABCD$, where the respective dimensions of A, B, C, D are 2×3, 3×4, 4×3 and 3×2, requires the same number of scalar multiplications.
   (e) (3 pts) True or false (justify). An articulation point of an undirected graph is defined as a node at which a depth-first search of the graph turns back.
   (f) (3 pts) True or false (justify). There is a $\frac{1}{2}$-correct Monte Carlo algorithm that decides in constant time whether an integer m is prime.
   (g) (3 pts) What is the probability of obtaining 2 successes in 3 attempts if each attempt succeeds with probability $\frac{3}{5}$?
   (h) (3 pts) True or false (justify). There is a $\frac{1}{m}$-correct "true-biased" Monte Carlo algorithm that decides in polynomial time whether an integer $m > 0$ is composite.
   (i) (3 pts) True or false (justify). It is possible to compute the OR of n bits in constant time using n processors hooked up to a central memory that operate in ERCW mode ("exclusive read concurrent write").

2. **(18 points)** Consider the following problem, called PAIR in French :

   <u>EVEN</u>
   DONNÉE: integer $m \geq 0$
   DÉCIDER: if m is even, i.e, divisible by 2 without remainder.

   A programming language has instructions to add 1 or to subtract 1 but has no other arithmetic (hence no $+, -, \times, //, \%$, nor binary shifts). As well, *for* loops only increment the loop index by 1 each time around the loop. The language otherwise allows initializing a variable to a constant, testing if two variables are equal, etc.

   (a) (6 pts) Using this language, give an algorithm $A(m)$ solving EVEN.
   (b) (3 pts) Estimate the exact order of the number $t_A(n)$ of instructions executed by $A(m)$ as a function of the number n of bits of m.
   (c) (6 pts) Now give in the same language a divide-and-conquer algorithm $B(m)$ solving EVEN. (No need for anything complicated.)
   (d) (3 pts) Write down the precise recurrence that describes the number $t_B(m)$ of instructions executed by your algorithm $B(m)$ as a function of m.

3. **(18 points)** Recall the KNAPSACK problem, SAC À DOS in French,

   Given : capacity $W \in \mathbb{R}^{\geq 0}$ and objects $1, 2, \ldots, n$ having weights $w_1, \ldots, w_n \in \mathbb{R}^{\geq 0}$ and values $v_1, \ldots, v_n \in \mathbb{R}^{\geq 0}$
   Compute : subset of objects of maximal value having total weight $\leq W$ (an object may not be picked more than once),

of which here is an instance :

$W = 10$	object 1	object 2	object 3	object 4	object 5
weights	1	2	4	4	4
values	3	1	4	5	2

(a) (2 pts) Is this problem solvable in general by a greedy algorithm ? YES NO

(b) (6 pts) Solve the above instance using the greedy approach.
(If you have answered NO above, then *before you begin, clearly describe* a KNAPSACK problem variant that your greedy technique will solve with certainty.)

What maximal value did you find ? $\boxed{}$

(c) (3 pts) What identity is at the heart of a dynamic programming strategy applied to the KNAPSACK problem ?

(d) (7 pts) Solve the above instance (the same as before) using dynamic programming, indicating the order and the steps used to fill an appropriate table.

What maximal value did you find ? $\boxed{}$

Which elements were chosen ?

4. **(17 points)** Let G be a directed graph, with positive lengths assigned to its edges. For every vertex v in G, we define the excentricity of v (denoted $E(v)$) as :

$$E(v) = \max_{w \in G}\{\ l(w,v) \mid l(w,v) \text{ the minimal length of a path from } w \text{ to } v\}$$

A vertex with minimum excentricity is called a center of G.

(a) (14 pts) Give the details of an algorithm (if you use an algorithm seen in class, then you must spell out this algorithm in full), in which dynamic programming is used, that computes a center of G.

(b) (3 pts) Give without justification the worst case complexity of your algorithm as a function of the number of vertices.

5. **(17 points)** The graph coloring problem is the following :

COLORIAGE

DONNÉE: adjacency matrix M of an m-vertex undirected graph
DÉCIDER: if there exists a vector $v \in \{\text{bleu, rouge, vert}\}^m$ such that for every edge (a,b) in the graph, $v[a] \neq v[b]$.

(a) (14 pts) Sketch a backtracking algorithm colo(M) for COLORIAGE.
Hint. Think of what might be a k-promising vector, $0 \leq k \leq m$.

(b) (3 pts) Even if no deterministic polynomial time algorithm solving COLORIAGE is known, could you turn your algorithm above into a polynomial time Las Vegas algorithm ? (Explain.)

BONNE CHANCE ! GOOD LUCK !