Examen 1 du cours de L3 Algorithmique et Programmation

Claire Mathieu et Jacques Stern

3 décembre 2012

Exercice 1. Calcul de températures moyennes [$\sim$ 3pts].

Le but de l'exercice est de concevoir une structure de données pour stocker des mesures de températures. Une *mesure* est de la forme (t, d) où t, un nombre réel, est une température, et où d, un entier, est le nombre de jours écoulés entre le 31 décembre 1999 et le jour de la mesure. Il peut y avoir plusieurs mesures un même jour, et il peut également n'y en avoir aucune. La structure de données doit pouvoir stocker l'ensemble S des mesures et en ajouter à S. On souhaite également faire des calculs de moyennes. Ainsi, la structure de données doit pouvoir gérer les opérations suivantes :

- *insérer*(t, d) qui ajoute à S une mesure supplémentaire, la mesure (t, d) ; et
- *moyenne*(d_1, d_2) où d_1 et d_2 sont deux dates telles que $d_2 \geq d_1$. Cette requête donne en résultat la moyenne uniforme des températures de tous les (t, d) de S tels que $d_1 \leq d \leq d_2$.

1. Montrer brièvement comment, avec une structure de données naïve, on peut obtenir une complexité de $O(1)$ pour chaque insertion et $O(n)$ pour chaque moyenne, où n est la taille de S.
2. Montrer comment, avec une autre méthode, on peut obtenir une complexité de $O(D)$ pour chaque insertion et $O(1)$ pour chaque requête de moyenne, où D est le nombre total de jours écoulés entre le 31 décembre 1999 et aujourd'hui.
3. Proposer une structure de données avec laquelle chaque insertion a complexité $O(\log n)$ et chaque requête de moyenne a complexité $O(\log n)$.

Exercice 2. Recoloriage de mot [$\sim$ 3pts].

Etant donné un ensemble C de couleurs, un entier n, on dit qu'une fonction $f : \{1, 2, \ldots, n\} \to C$ est *convexe* si les positions qui ont une même couleur sont consécutives : si $f(i) = f(j) = c$ avec $i \leq j$, alors pour tout $k \in [i, j]$, on a $f(k) = c$.
Etant donné f non convexe, on dit qu'une couleur $c \in C$ est *problématique* s'il existe des positions $i < k < j$ telles que $f(i) = f(j) = c \neq f(k)$.
Soit f une fonction et ℓ le nombre de couleurs problématiques pour f.

1. Donner un algorithme en $O(2^\ell n^2)$ qui donne en sortie le nombre minimum de valeurs de f à modifier pour rendre f convexe.
2. Le modifier afin qu'il donne en sortie une fonction g convexe minimisant le nombre de i tels que $f(i) \neq g(i)$.

Exercice 3. Mot sans répétition [$\sim$ 3pts].

1. Etant donné un mot $w = w_1 w_2 \ldots w_n$ sur un alphabet A, donner un algorithme pour trouver la longueur maximale d'un sous-mot dont les lettres sont toutes distinctes. Par exemple, si $w =$ "$algorithmique$" alors la longueur maximale est 8, pour le sous-mot $algorithm$. Votre algorithme doit avoir complexité $O(n|A|)$.
2. Etant donné une suite d'*entiers* $w = w_1 w_2 \ldots w_n$, donner un algorithme pour trouver la longueur maximale d'un sous-mot dont les *entiers* sont tous distincts. Par exemple, si $w = (3, 54, 2, 3, 5, 128, 7, 3)$ alors la longueur maximale est 4, pour le sous-mot $(5, 128, 7, 3)$. Votre algorithme doit avoir complexité $O(n^3)$ ou mieux.

Exercice 4. Chemin hamiltonien dans un tournoi [$\sim$ 3pts].

Soit G un tournoi, c'est-à-dire un graphe orienté tel que pour toute paire de sommets u, v, exactement un des deux arcs (u, v) et (v, u) est dans E.

1. Donner un algorithme qui calcule un chemin hamiltonien de G, c'est-à-dire un chemin consistant en $n - 1$ arcs de G et visitant tous les sommets de G exactement une fois. (n est le nombre de sommets de G.)
2. Montrer que G ne possède pas nécessairement de circuit hamiltonien, c'est-à-dire de cycle consistant en n arcs de G et visitant tous les sommets de G exactement une fois.

Exercice 5. Coloriage de graphe [$\sim$ 3pts].

Soit G un graphe non-orienté. Un *coloriage* de G assigne une couleur à chaque sommet de façon que toute paire de sommets voisins aient des couleurs distinctes.

1. Montrer que tout arbre T peut être colorié avec au plus 2 couleurs. Donner un algorithme coloriant T avec au plus deux couleurs.
2. Soit $G = T_1 \cup T_2 \cup T_3$ un graphe formé de l'union de trois arbres, tous trois sur un même ensemble de sommets : l'arête $\{u, v\}$ est dans G si et seulement si elle est présente dans au moins un des trois arbres. Montrer que G est coloriable avec au plus 8 couleurs.
3. Montrer qu'en fait, $G = T_1 \cup T_2 \cup T_3$ est coloriable avec au plus 6 couleurs. Donner un algorithme de coloriage de G avec au plus 6 couleurs.
4. Est-il toujours possible de colorier $G = T_1 \cup T_2 \cup T_3$ avec 5 couleurs ?
5. Généraliser à $G = T_1 \cup T_2 \cup \cdots \cup T_k$.

Exercice 6. Acteurs et actrices [$\sim$ 3pts].

On considère un ensemble X de n acteurs et un ensemble Y de n actrices, ainsi qu'une matrice M telle que $M[x, y]$ est un booléen qui est vrai si et seulement si l'acteur x et l'actrice y ont joué ensemble dans un même film. Deux joueurs P_X et P_Y jouent à un jeu. P_X nomme un acteur x_1 de X, puis P_Y une actrice y_1 de Y telle que $M[x_1, y_1]$ soit vrai, puis P_X un acteur $x_2 \neq x_1$ tel que $P[x_2, y_1]$ soit vrai, puis P_Y une actrice $y_2 \neq y_1$ telle que $M[x_2, y_2]$ soit vrai, et ainsi de suite pour construire une suite $(x_1, y_1, x_2, y_2, x_3, y_3, \ldots)$. Le jeu s'arrête quand un joueur ne peut pas (sans citer quelqu'un déjà nommé précédemment) nommer un acteur ou une actrice qui a joué avec la dernière actrice ou le dernier acteur mentionné. Ce joueur a alors perdu.

1. Donner un algorithme qui détermine lequel des deux joueurs va gagner.
2. Donner un algorithme qui détermine une stratégie gagnante pour ce joueur.

Exercice 7. Arbre couvrant maximum [$\sim$ 3pts].

Donner un algorithme de complexité linéaire $O(m + n)$ qui étant donné un graphe non-orienté connexe $G = (V, E)$ avec des poids sur les arêtes, $w_e \geq 0$, construit un arbre couvrant de G qui minimise le poids de l'*arête de poids maximum*. (Indication : on rappelle qu'on peut calculer le médian d'un ensemble de nombres en temps linéaire.)