ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 11 7 Décembre 2007

1. L'algorithme de Karp

On considère l'algorithme de Karp sur un graphe $G = (V, E)$ avec poids $w : E \to \mathbb{R}$.

Les points a) et b) ci-dessous sont l'exercice qui se trouve au milieu de la page 196 du polycopié.

a) Montrer que l'égalité suivante est vraie :

$$F_k(x) = \min_{u \in V \wedge (u,x) \in E} \{F_{k-1}(u) + w(u, x)\},$$

ou $F_k(\cdot)$ est (par définition) le poids du plus léger chemin de s à x de longueur k. (Voir le cours pour les notations en détail.)

b) Montrer que les valeurs de $F_k, F_{k-1}, \cdots, F_0$ peuvent être calculées en $O(|E| \cdot (k + 1))$ opérations.

c) Ecrire explicitement (en pseudo-code) l'algorithme de Karp.

2.Shift cyclique

Supposons que $a = (a_0, a_1, \ldots, a_{n-1})$ est un shift cyclique d'une suite triée dont tous les éléments sont distincts. (Par exemple, $(6, 7, 11, 18, 1, 2)$ est de cette forme, puisqu'on peut shifter ses composantes de quatre positions vers la gauche afin d'obtenir $(1, 2, 6, 7, 11, 18)$, qui est triée.)

a) Soient a_i et a_j, avec $i < j$. Soit k la position du plus petit élément de la suite. Montrer que

$$a_j < a_i \iff i < k \leq j.$$

b) Donner un algorithme (en pseudo-code) qui identifie en $O(\log(n))$ opérations le shift à effectuer pour obtenir une suite triée.

c) Montrer que n'importe quel algorithme qui résout ce problème en se basant sur la comparaison de clés est $\Omega(\log(n))$.

3.Induction et Récursion

Dans cet exercice nous n'écrivons pas de virgules dans les vecteurs binaires, i.e. 0101 dénote le vecteur $(0, 1, 0, 1)$. Nous dénotons par ":" l'opérateur de concaténation. Par exemple $1111 : 000 = 1111000$ et $101 : 1101 = 1011101$.

Nous définissons l'opérateur Γ agissant sur des mots de vecteurs binaires comme suit : si $x_1, x_2, x_3, \ldots, x_n$ sont des vecteurs binaires, alors on définit

$$\Gamma(x_1, x_2, x_3, \ldots, x_n) := (0 : x_1, 0 : x_2, \ldots, 0 : x_n, 1 : x_n, 1 : x_{n-1}, 1 : x_{n-2}, \ldots, 1 : x_1).$$

Par exemple, on a $\Gamma(010, 101) = (0010, 0101, 1101, 1010)$.

On définit la suite $(A_i)_{i\in\mathbb{N}}$ de vecteurs binaires comme suit :

$$\begin{aligned} A_1 &= (0,1), \\ A_i &= \Gamma(A_{i-1}) \qquad \text{pour tout } i > 1. \end{aligned}$$

Donc par exemple,

$$\begin{aligned} A_2 &= (00, 01, 11, 10), \\ \text{et } A_3 &= (000, 001, 011, 010, 110, 111, 101, 100). \end{aligned}$$

a) Donner A_4.

b) Montrer que les A_i vérifient la propriété que deux éléments successifs diffèrent en exactement une position. (Par exemple, pour A_2, on voit que 00 et 01 ne diffèrent qu'à la deuxième position, 01 et 11 ne diffèrent qu'à la première position, et 11 et 10 ne diffèrent qu'à la deuxième position.)

c) Montrer que A_n contient chacun des 2^n n-tuples binaires possibles exactement une fois.

4. Matching

Le matching suivant est-il maximal ? Justifier. (Les arêtes en gras sont celles qui forment le matching.)