Université de Marne-la-Vallée

Licences d'informatique et d'I.U.P.
Examen d'algorithmique

Mai 2003, durée : 2h

Seule la consultation des supports de cours, travaux dirigés et travaux pratiques est autorisée pendant le contrôle.

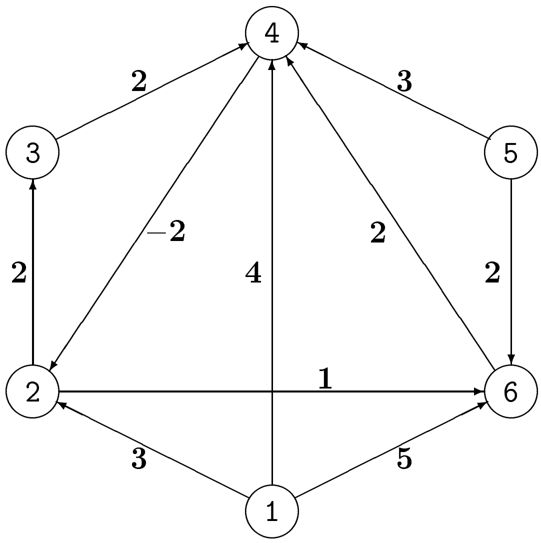
Distances

On considère dans cet exercice les coûts minimaux des chemins entre les sommets s_1, s_2, ..., s_n d'un graphe orienté $\mathcal{G}$. Les réponses aux questions **2** et **3** doivent être fournies sur les pages 5 et 6.

1) Donner la récurrence utilisée par l'algorithme de Floyd pour calculer les coûts minimaux. Quand $\mathcal{G}$ est fortement connexe, montrer que l'un des coûts est égal à $-\infty$ si et seulement s'il y a un cycle de coût négatif.

2) Calculer les coûts à chaque étape de l'algorithme de Floyd pour le graphe ci-dessous (tableaux C_0 à C_6 des pages 5 et 6).

3) Indiquer les prédécesseurs des sommets (tableaux P_0 à P_6 des pages 5 et 6).



Classes

On considère des classes d'équivalence d'entiers compris entre 0 et $n-1$ sur lesquelles on veut appliquer des opérations d'union. Un ensemble de classes (qui constitue une partition de l'ensemble $\{0, 1, \ldots, n-1\}$) est représenté par une forêt d'arbres ordinaires (les nœuds de chaque arbre sont les éléments de la classe associée) au moyen de deux tableaux P (parents ou pères) et T (tailles) : pour chaque $i = 0, 1, \ldots, n-1$,

$$P[i] = \begin{cases} \text{parent de } i & \text{si } P[i] \geq 0, \\ -c & \text{si } P[i] < 0. \end{cases}$$

Dans le second cas, c est le numéro de la classe de i et $T[c]$ est son nombre d'éléments.

4) Dessiner la forêt correspondant aux tables :

	0	1	2	3	4	5	6	7	8
P	2	4	-1	8	-2	3	4	4	-3

	1	2	3
T	2	4	3

5) Décrire des algorithmes RACINE(i), CLASSE(i) et TAILLE(i), pour calculer respectivement la racine de l'arbre contenant l'entier i, le numéro de la classe de i, et la taille de la classe de i. Indiquer leur temps d'exécution.

6) Décrire un algorithme qui effectue l'union des classes de deux entiers i et j ($0 \leq i < n$, $0 \leq j < n$) et qui s'exécute en temps constant après le calcul de RACINE(i) et RACINE(j).

7) Décrire une stratégie qui, lorsque k unions sont appliquées à des arbres de hauteur maximale h, permet d'obtenir des arbres dont la hauteur ne dépasse pas $h + \log k$.

Alignement

Un alignement entre deux mots $x = x[0]x[1]\cdots x[m-1]$ et $y = y[0]y[1]\cdots y[n-1]$ sur l'alphabet A est un tableau z de la forme

$$\begin{array}{cccc} u_0 & u_1 & \cdots & u_{p-1} \\ v_0 & v_1 & \cdots & v_{p-1} \end{array}$$

où les u_i et v_i sont des lettres de A ou le mot vide ε, mais u_i et v_i ne sont pas simultanément vides, et $x = u_0u_1\cdots u_{p-1}$, $y = v_0v_1\cdots v_{p-1}$.

Le coût de l'alignement z, noté *coût*(z), est la somme $\sum_{i=0}^{p-1} c\begin{pmatrix} u_i \\ v_i \end{pmatrix}$

où $c\begin{pmatrix} \mathtt{A} \\ \mathtt{A} \end{pmatrix} = 0$, $c\begin{pmatrix} \mathtt{A} \\ \mathtt{B} \end{pmatrix} = 3$, $c\begin{pmatrix} \mathtt{A} \\ \varepsilon \end{pmatrix} = 1$, $c\begin{pmatrix} \varepsilon \\ \mathtt{B} \end{pmatrix} = 1$, pour des lettres $\mathtt{A} \neq \mathtt{B}$.

Par exemple,

$$\begin{array}{cccccc} \mathtt{A} & \mathtt{B} & \mathtt{B} & \mathtt{B} & \varepsilon & \varepsilon \\ \mathtt{A} & \varepsilon & \mathtt{A} & \mathtt{B} & \mathtt{A} & \mathtt{A} \end{array}$$

est un alignement des mots `ABBB` et `AABAA`. Son coût est $0+1+3+0+1+1 = 6$.

On s'intéresse au calcul de $L(x,y) = \min\{\mathit{co\hat{u}t}(z) \mid z \text{ alignement de } x \text{ et } y\}$.

Soit T la matrice définie à partir de x et y par :

$$\begin{array}{rcl} T[-1,-1] & = & 0 \\ T[i,-1] & = & T[i-1,-1]+1 \\ T[-1,j] & = & T[-1,j-1]+1 \\ T[i,j] & = & \min\begin{cases} T[i-1,j-1] + \mathit{Sub}(x[i],y[j]) \\ T[i-1,j]+1 \\ T[i,j-1]+1 \end{cases} \end{array}$$

où $\mathit{Sub}(x[i],y[j]) = 0$ si $x[i] = y[j]$ et $\mathit{Sub}(x[i],y[j]) = 3$ si $x[i] \neq y[j]$, pour $i = 0,1,\ldots,m-1$ et $j = 0,1,\ldots,n-1$.

8) Montrer l'égalité $T[i,j] = L(x[0]x[1]\cdots x[i], y[0]y[1]\cdots y[j])$, pour $i = 0,1,\ldots,m-1$ et $j = 0,1,\ldots,n-1$.

9) Calculer dans le tableau de la page 4 la matrice T correspondant aux deux mots $r =$ `EAWACQGKL` et $s =$ `ERDAWCQPGKWY`, et donner la valeur de $L(r,s)$.

10) Indiquer comment calculer un alignement dont le coût est $L(x,y)$ à partir de la matrice T, et tracer le résultat pour r et s sur le tableau de la page 4.

Alignement (questions **9** et **10**).

T	j	-1	0	1	2	3	4	5	6	7	8	9	10	11
i		$y[j]$	E	R	D	A	W	C	Q	P	G	K	W	Y
-1	$x[i]$													
0	E													
1	A													
2	W													
3	A													
4	C													
5	Q													
6	G													
7	K													
8	L													

Distances (questions **2** et **3**).

Note : indiquer les valeurs modifiées à chaque étape du calcul.

C_0	1	2	3	4	5	6
1	0	3	∞	4	∞	5
2	∞	0	2	∞	∞	1
3	∞	∞	0	2	∞	∞
4	∞	-2	∞	0	∞	∞
5	∞	∞	∞	3	0	2
6	∞	∞	∞	2	∞	0

P_0	1	2	3	4	5	6
1	–	1	–	1	–	1
2	–	–	2	–	–	2
3	–	–	–	3	–	–
4	–	4	–	–	–	–
5	–	–	–	5	–	5
6	–	–	–	6	–	–

C_1	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

P_1	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

C_2	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

P_2	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

C_3	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

P_3	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

C_4	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

P_4	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

C_5	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

P_5	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

C_6	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

P_6	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						