L3-INFO Algorithmique et Modélisation

Université Joseph Fourier GRENOBLE UFR IM²AG

Quick – 7 avril 2015 – durée 1 h

Sont interdits : les documents, les ordinateurs, les téléphones (incluant smartphone, tablettes,... tout ce qui contient un dispositif électronique).
Seuls les dictionnaires papier pour les personnes de langue étrangère sont autorisés.

La clarté des raisonnements et la qualité de la rédaction interviendront pour une part importante dans l'appréciation des copies.

Exercice 1 : Complexité récursive

Pour un problème donné, un algorithme naïf a une complexité en $O(n^3)$. On a aussi trouvé deux solutions de type diviser pour régner :

Div1 Découper le problème de taille n en deux sous-problèmes de taille $n/2$ et les recombiner en temps $O(n^2)$.

Div2 Découper le problème en quatre sous-problèmes de taille $n/3$ et les recombiner en temps $O(n)$.

1. Quelle est la complexité de Div1 ?
2. Quelle est la complexité de Div2 ?
3. Quelle solution choisir : naïf, div1 ou div2 ?

Exercice 2 : Programmation dynamique

On se donne une matrice C de taille $n \times m$ (n et m sont deux entiers supérieurs ou égaux à 1). On chercher à calculer la quantité $D_{n,m}$, définie par récurrence par la formule suivante :

$$D_{i,j} = \begin{cases} i \text{ si } j = 0 \\ j \text{ si } i = 0 \\ \min(D_{i-1,j} + 1, D_{i,j-1} + 1, D_{i-1,j-1} + C_{ij}) \text{ sinon} \end{cases}$$

1. Écrire un algorithme récursif qui prend en entrée une matrice C de taille $n \times m$ et calcule $D_{n,m}$. Quelle est sa complexité ?
2. La relation de récurrence précédente se prête particulièrement bien à l'expression d'une implémentation par programmation dynamique. Écrire l'algorithme correspondant à cette solution. Quelle est la complexité de votre algorithme ?

Exercice 3 : Plus courts chemins

On se donne un graphe orienté (S, A) ayant $|S| = n$ sommets et $|A| = m$ arcs. Soit $o \in S$ un sommet de ce graphe. On cherche à calculer l'ensemble des sommets pour lesquels il existe un chemin partant de o. Pour cela, on utilise la propriété suivante :

(P) il existe un chemin de o à j de longueur inférieure ou égale à k si :

- il existe un chemin de o à j de longueur inférieure ou égale à $k-1$;

ou

- il existe un arc $(i, j) \in A$ et un chemin de o à i de longueur inférieure ou égale à $k-1$.

On note $C_{j,k}$ une variable qui vaut *vrai* s'il existe un chemin de o à j de longueur inférieure ou égale à k et qui vaut faux sinon.

1. Formaliser une relation de récurrence sur $C_{j,k}$ utilisant la propriété (P).
2. La relation de récurrence précédente se prête particulièrement bien à l'expression d'une implémentation par programmation dynamique. Écrire l'algorithme correspondant à cette solution. L'algorithme prendra en entrée un sommet o et l'ensemble des arcs A et rendra en sortie un vecteur C tel que $C_i = vrai$ s'il existe un chemin entre o et i et $C_i = faux$ sinon.
3. Calculer la complexité de votre algorithme en fonction de n et m.
4. On associe à chaque arc $(i, j) \in A$ un poids $P_{ij} \geq 0$ et on cherche à calculer un chemin de poids minimal entre deux sommets o et d. Écrire un algorithme utilisant la propriété (P) qui affiche un chemin de poids minimal de o à d.