Algorithmique — M1

Partiel du 16 novembre 2010

Université Paris Diderot

Documents autorisés : une feuille de papier format A4.
Durée : 1h30 ; le sujet est recto-verso ; le barème est indicatif
De préférence, écrivez vos algorithmes en pseudo-code.

On applique les cours

Exercice 1 (1 point) – *Récurrence*

Étant donné que

$$T(n) = 27T(\lfloor n/3 \rfloor) + 3n^3 + \log n$$

trouvez le comportement asymptotique de $T(n)$. Justifiez votre réponse.

Exercice 2 (2 points) – *La science*

Un chercheur souhaite assister à un nombre maximum de conférences durant le mois de décembre. Les dates prévues sont :

Le colloque des savants 1-12 décembre.

Le colloque des ignorants 8-10 décembre.

La petite conférence sur les grands graphes 14-15 décembre.

La grande conférence sur les petits graphes 3-19 décembre.

La semaine des automates 18-25 décembre.

La journée des grammaires 11 décembre.

Le workshop de Noël : 24-31 décembre.

1. C'est une instance d'un problème vu en cours. Lequel ?
2. Expliquez l'algorithme de cours pour ce problème. (inutile de prouver sa correction)
3. Appliquez cet algorithme et trouvez la solution du problème.

On invente des algorithmes très simples

Exercice 3 (2 points) – *Méthode imposée pour un problème trivial*

Écrire une fonction qui compte le nombre d'occurrences d'un élément e dans un tableau $A[s..f]$.

1. Écrivez un algorithme de type diviser-pour-régner qui résout ce problème.
2. Analysez sa complexité.
3. Comparez-la avec celle de l'algorithme naïf vu en L1.

Exercice 4 (3 points) – *Encore un sac à dos (monotone)*

Nous avons n objets de poids $p[1] \leqslant p[2] \leqslant \cdots \leqslant p[n]$ et de valeur $v[1] \geqslant v[2] \geqslant \cdots \geqslant v[n]$. Il faut trouver un sous-ensemble des objets de poids total $\leqslant W$kg et de valeur maximale.

1. Écrivez un algorithme glouton qui résout ce problème.
2. Analysez sa complexité.
3. Justifiez sa correction.

On invente des algorithmes moins simples

Exercice 5 (6 points) – *Un problème NP-complet : chemin d'Hamilton*

Les n villes du pays Back-And-Track-Land sont reliés par un réseau routier. La matrice d'adjacence $M[i,j]$ (de taille $n \times n$) représente ce réseau de façon habituelle : $M[i,j] = 1$ s'il y a une route directe de la ville i vers la ville j ; et $M[i,j] = 0$ s'il n'y en a pas. Mr Hamilton, explorateur célèbre, veut visiter toutes les villes, en passant par chaque ville une seule fois [1]. Aidons-le à planifier son voyage.

Dans cet exercice il faut trouver un algorithme retour-arrière qui trouve un itinéraire si celui-ci existe. On va appeler une solution partielle un chemin de longueur k (qui utilise les routes existantes et ne revisite jamais une même ville). On représentera cette solution partielle par un tableau de numéros de villes $R = [i_1, \ldots, i_k]$.

1. Écrivez une fonction booléenne $\mathtt{test}(R, k, M, n)$ qui teste si le tableau $R[k]$ est une solution partielle pour un réseau routier représenté par un tableau (matrice d'adjacence) $M[n,n]$.
2. Quelles sont les solutions partielles de taille 1 ? Comment à partir d'une solution partielle de taille k passer à ses extensions de taille $k+1$? Comment détecter si on a déjà trouvé un chemin pour Mr Hamilton ?
3. Écrivez un algorithme retour-arrière de recherche de chemin d'Hamilton.
4. Estimez la complexité de votre algorithme.

Exercice 6 (6 points) – *Encore un cinéma : multiplex*

Demain dans le "Ciné Glouton" il y aura n séances et plusieurs salles. La séance numéro i commence à l'heure $d[i]$ et se termine à l'heure $f[i]$. Il faut planifier toutes les séances en utilisant le moins de salles possible.

1. Écrivez un algorithme qui résout le problème dans le cas général. On pourrait le faire dans l'ordre croissant de $d[i]$.
2. Analysez sa complexité.
3. Justifiez sa correction.

1. Ne pas confondre avec le chemin eulerien : Mr Euler veut parcourir *chaque route* une fois et une seule, tandis que Mr Hamilton veut visiter *chaque ville* une fois et une seule.