INF431

Programmation dynamique
Sujet proposé par Bruno Salvy
Version: 243:2470M

Comme vu en cours, la programmation dynamique permet de calculer des solutions optimales à des problèmes pour lesquels ces solutions peuvent se construire à partir de solutions optimales de sous-problèmes. L'exercice consiste alors à découvrir la bonne décomposition du problème donné.

1 Travaillez efficacement

La fin de l'année approche. Vous devez réviser m matières différentes ; ces matières sont désignées par les numéros de 1 à m.

Dans chaque matière, vous obtiendrez une note entière comprise entre 0 et g avec $g > 1$. Vous avez pu déterminer, pour chaque matière, une fonction qui vous donne votre note à partir du nombre d'heures de révision que vous consacrez à cette matière. Si vous travaillez la matière i pendant h heures, vous obtiendrez la note $p_i(h)$. Pour simplifier, on suppose que l'on ne peut pas diviser les heures, c'est-à-dire que h est entier. On peut supposer que les fonctions p_i sont croissantes.

Il ne vous reste plus que n heures avant les examens, et vous devez les répartir au mieux entre les matières. C'est-à-dire que vous devez trouver pour chaque matière i un nombre h_i d'heures tel que :

$$\sum_{i=1}^{m} h_i = n \qquad \text{et} \qquad \sum_{i=1}^{m} p_i(h_i) \quad \text{est maximal.}$$

Question 1 Écrire le pseudo-code d'un algorithme qui calcule le nombre maximal de points que vous pouvez obtenir. Donnez la complexité en temps de votre algorithme, qui doit être polynomiale en n et en m. $\diamond$

Question 2 A partir de là, écrivez le pseudo-code d'un algorithme qui calcule effectivement les h_i correspondant à la répartition optimale entre les matières. $\diamond$

2 Plus longue sous-séquence commune

Une mesure de similarité entre deux séquences d'ADN est la longueur de leur plus longue sous-séquence commune. Précisément, si les deux séquences sont $x_1, \dots, x_n$ et $y_1, \dots, y_m$ (où les x_i et y_i appartiennent à l'alphabet $\{A, C, G, T\}$), il s'agit de trouver l'entier k maximal tel qu'existent deux suites d'indices $j_0 < j_1 < \cdots < j_{k-1}$ et $\ell_0 < \ell_1 < \cdots < \ell_{k-1}$ avec $x_{j_i} = y_{\ell_i}$ pour tout $0 \le i < k$.

Question 3 Écrire le pseudo-code d'une solution par programmation dynamique en temps $\mathrm{O}(nm)$ au problème de la plus longue sous-séquence commune. $\diamond$

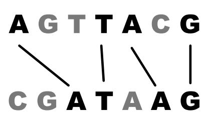

FIGURE 1 – Une plus longue sous-séquence commune à A,G,T,T,A,C,G et C,G,A,T,A,A,G est A,T,A,G.

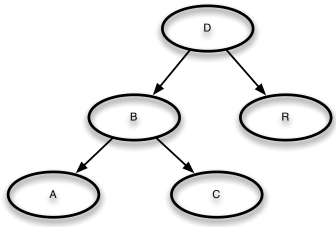

FIGURE 2 – Arbre binaire de recherche

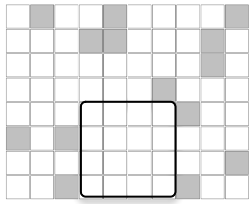

FIGURE 3 – Un plus grand sous-carré monochromatique sur une matrice à deux symboles *blanc* et *noir*.

3 Arbres de recherche binaires optimaux

Un arbre binaire de recherche est une structure de données pour des clés ordonnées. Il s'agit d'un arbre binaire, dont les nœuds contiennent des clés. Les clés inférieures à celle de la racine sont stockées dans des nœuds de son sous-arbre gauche, alors que les clés supérieures sont stockées dans son sous-arbre droit, la même règle étant appliquée récursivement (voir l'exemple Fig. 2 où les clés sont des lettres). Tester la présence d'une clé est très simple : soit elle est à la racine et la recherche est terminée, soit elle est inférieure à la clef de la racine et il faut la chercher récursivement dans le sous-arbre gauche, soit enfin il faut la chercher dans le sous-arbre droit.

Si un arbre binaire de recherche est utilisé pour de la vérification orthographique, avec les mots pour clés, il est avantageux de tenir compte des fréquences des mots dans la langue pour faire apparaître les plus fréquents vers la racine. Plus précisément, étant données n clés $x_1 < \cdots < x_n$ et leurs fréquences $f_1, \ldots, f_n$, il s'agit de construire un arbre binaire de recherche sur ces clés qui minimise le coût $f_1 d_1 + \cdots + f_n d_n$, où d_i représente la profondeur du nœud contenant la clé x_i (la profondeur de la racine étant 1).

Question 4 Écrire le pseudo-code d'une procédure calculant par programmation dynamique le coût de l'arbre binaire optimal, et estimer sa complexité. [Indication : les clés appartenant à un même sous-arbre sont consécutives.] ◇

4 Plus grand sous-carré monochromatique

Étant donné un tableau M de dimension $n \times m$ dont les entrées sont des couleurs (représentées par des entiers), un sous-carré monochromatique de côté k est donné par $i \leq n - k$ et $j \leq m - k$ tels que tous les $M[a, b]$ sont égaux pour $i \leq a < i + k$, $j \leq b < j + k$.

Question 5 Écrire le pseudo-code d'une procédure qui, étant donné M, renvoie un plus grand sous-carré monochromatique. Estimer sa complexité. ◇