

ALGORITHMIQUE 2, TD n° 8

DEVAN SOHIER

1. PLUS LONGUE SOUS-SÉQUENCE COMMUNE

On s'intéresse au problème de la recherche d'une plus longue sous-séquence commune à deux mots. Etant donné un mot ω , on appelle sous-séquence de ω un mot $\omega_{j_1}\omega_{j_2}\dots\omega_{j_k}$ avec $j_1 < j_2 < \dots < j_k$. Notons $s(\omega, \omega')$ la plus longue sous-séquence commune à ω et ω' , c'est-à-dire le mot le plus long qui soit une sous-séquence de ω et une sous-séquence de ω' .

Que valent $s(\omega, \epsilon)$ et $s(\epsilon, \omega)$.

Etant donnés deux mots ω et ω' , et deux caractères α et β (non-nécessairement distincts), exprimer $s(\omega\alpha, \omega'\beta)$ en fonction de $s(\omega\alpha, \omega')$, $s(\omega, \omega'\beta)$ et $s(\omega, \omega')$. Discutez de l'interprétation de cette formule comme propriété de sous-structure optimale.

Déduisez en un algorithme calculant la plus longue sous-séquence commune à deux mots : cet algorithme consistera à remplir un tableau dont les lignes et les colonnes sont indexés par les caractères des deux mots, et dont la case (i, j) contiendra la longueur de la plus longue sous-séquence commune aux préfixes de longueurs i et j des deux mots.

Adaptez cet algorithme pour qu'il calcule le *diff* entre deux textes.

2. LE PROBLÈME DU SAC À DOS

Un randonneur, partant pour une longue excursion, détermine avec soin le contenu de son sac à dos. Compte tenu des équipements indispensables déjà chargés, le poids total de nourriture emportée ne devra pas excéder 16 kilos. Il dispose, en quantités limitées, de trois types d'aliments, de valeur nutritive variable, dont les poids unitaires sont différents ; les aliments sont conditionnés par unités non fractionnables.

Aliment	I	II	III
Poids unitaires (en Kg)	7	5	2
quantités disponibles	4	3	4
valeurs nutritives	15	10	4

Le randonneur cherche la quantité de chaque aliment à emporter, de façon à maximiser la valeur nutritive totale, tout en tenant compte de la limite de 16 kg qu'il s'est fixée.

Il désire... utiliser la programmation dynamique pour résoudre ce problème dit du "knapsack". D'une manière générale, soient :

- : n , le nombre d'aliments différents
- : p_i , le poids unitaire de l'aliment i
- : q_i , le nombre d'unités disponibles de l'aliment i
- : c_i , sa valeur nutritive

- : x_i , la quantité de l'aliment i emportée (en nombre d'unités)
- : P ; le poids maximal de nourriture qu'il est possible d'emporter.

- (1) Donner une formalisation du problème en tant que programme linéaire en nombres entiers.
- (2) Noter $f_i(Q)$ la valeur nutritive maximale du sac à dos chargé (optimalement) avec uniquement les i premiers produits, et de poids total Q ($Q \preceq P$).
Donner les formules d'optimisation séquentielle.
- (3) Résoudre numériquement