Fiche TD 3 - Programmation dynamique

Exercice 1 - Triangulation de polygones

On considère les polygones convexes du plan. Une triangulation d'un polygone est un ensemble de cordes qui ne se coupent pas à l'intérieur du polygone et qui le divisent en triangles.

1 - Montrer qu'une triangulation d'un polygone à n côtés a $(n-3)$ cordes et $(n-2)$ triangles.

2 - Le problème est celui de la triangulation optimale de polygones. On part d'un polygone convexe $P = \langle v_0, ..., v_n \rangle$, où $v_0, \ldots, v_n$ sont les sommets du polygone donnés dans l'ordre direct, et d'une fonction de pondération w définie sur les triangles formés par les côtés et les cordes de P (par exemple $w(i,j,k) = \|v_i v_j\| + \|v_j v_k\| + \|v_k v_i\|$ est le périmètre du triangle $v_i v_j v_k$). Le problème est de trouver une triangulation qui minimise la somme des poids des triangles de la triangulation.

On définit pour $1 \leq i < j \leq n$, $t[i,j]$ comme la pondération d'une triangulation optimale du polygone $\langle v_{i-1}, ..., v_j \rangle$, avec $t[i,i] = 0$ pour tout $1 \leq i \leq n$.

Définir t récursivement, en déduire un algorithme et sa complexité.

3 - Si la fonction de poids est quelconque, combien faut-il de valeurs pour la définir sur tout triangle du polygone ? Comparez avec la complexité obtenue.

4 - Si le poids d'un triangle est égal à son aire, que pensez-vous de l'algorithme que vous avez proposé ?

Exercice 2 - Jeu de construction

On veut construire une tour la plus haute possible à partir de différentes briques. On dispose de n types de briques et d'un nombre illimité de briques de chaque type. Chaque brique de type i est un parallélépipède de taille (x_i, y_i, z_i) et peut être orientée dans tous les sens, deux dimensions formant la base et la troisième dimension formant la hauteur.

Dans la construction de la tour, une brique ne peut être placée au dessus d'une autre que si les deux dimensions de la base de la brique du dessus sont *strictement inférieures* aux dimensions de la base de la brique du dessous.

Proposer un algorithme efficace pour construire une tour de hauteur maximale.

Exercice 3 - Impression équilibrée

Le problème est l'impression équilibrée d'un paragraphe sur une imprimante. Le texte d'entrée est une séquence de n mots de longueurs l_1,l_2,...,l_n (mesurées en caractères). On souhaite imprimer ce paragraphe de manière équilibrée sur un certain nombre de lignes qui contiennent un maximum de M caractères chacune. Le critère d'*équilibre* est le suivant.

Si une ligne donnée contient les mots i à j (avec $i \leq j$) et qu'on laisse exactement un espace entre deux mots, le nombre de caractères d'espacement supplémentaires à la fin de la ligne est $M - j + i - \sum_{k=i}^{j} l_k$, qui doit être positif ou nul pour que les mots tiennent sur la ligne. L'objectif est de minimiser la somme, sur toutes les lignes *hormis la dernière*, des cubes des nombres de caractères d'espacement présents à la fin de chaque ligne.

1 - Est-ce que l'algorithme glouton consistant à remplir les lignes une à une en mettant à chaque fois le maximum de mots possibles sur la ligne en cours, fournit l'optimum ?

2 - Donner un algorithme de programmation dynamique résolvant le problème. Analyser sa complexité en temps et en espace.

3 - Supposons que pour la fonction de coût à minimiser, on ait simplement choisi la somme des nombres de caractères d'espacement présents à la fin de chaque ligne. Est-ce que l'on peut faire mieux en complexité que pour la question 2 ?

4 - (*Plus informel*) Qu'est-ce qui à votre avis peut justifier le choix de prendre les cubes plutôt que simplement les nombres de caractères d'espacement en fin de ligne ?