TD 1 : Diviser pour régner

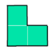
5 Septembre 2024

Coordonnées des chargés de TD : `malo.revel@ens-rennes.fr` `isseinie.calviac@irisa.fr`
Les TDs ont lieu les jeudis de 9h45 à 11h15, se fier à ADE pour la salle.

1 Pavage avec des triominos

On considère le problème suivant :

— **Entrée :** un carré de $2^n \times 2^n$ cases et une case particulière T ;

— **Sortie :** le pavage du carré, privé de la case T à l'aide de triominos de la forme  .

QUESTION 1 – Proposer un algorithme de type « diviser pour régner » résolvant le problème. Quelle est sa complexité en fonction de n ?

2 Théorème fondamental sur la complexité

Théorème 1. *Soit $a \geq 1$, $b \geq 2$ et $d \geq 0$. Considérons une suite $(\tau(n))_{n\in\mathbb{N}}$ qui vérifie, pour tout $n \geq 2$,*

$$\tau(n) = a\tau\left(\lceil\frac{n}{b}\rceil\right) + \mathcal{O}(n^d).$$

Alors :

— *Si $a < b^d$, alors $\tau(n) = \mathcal{O}(n^d)$*

— *Si $a = b^d$, alors $\tau(n) = \mathcal{O}(n^d \log_b n)$*

— *Si $a > b^d$, alors $\tau(n) = \mathcal{O}(n^{\log_b a})$*

QUESTION 2 – Exprimer la complexité de l'algorithme précédent en se servant du théorème fondamental.

3 Algorithme de Karatsuba

On considère le problème suivant :

— **Entrée :** deux nombres entiers x et y avec n chiffres chacun (en base 10) ;

— **Sortie :** la valeur de $x \times y$.

QUESTION 3 – Quelle est la complexité de l'algorithme naïf en fonction de n (en supposant qu'une multiplication par 10^k se fait en $O(1)$) ?

On remarque que pour tout k, en décomposant $x = (a \times 10^k + b)$ et $y = (c \times 10^k + d)$, on a

$$\begin{aligned}(a \times 10^k + b)(c \times 10^k + d) &= ac \times 10^{2k} + (ad + bc) \times 10^k + bd \\ &= ac \times 10^{2k} + (ac + bd - (a - b)(c - d)) \times 10^k + bd\end{aligned}$$

QUESTION 4 – En déduire un algorithme de type « diviser pour régner » résolvant le problème et donner sa complexité en fonction de n.

4 Silhouette d'une ville

On cherche à calculer la silhouette d'une ville constituée d'un groupe d'immeubles, vue (en 2 dimensions) par un observateur distant. Chaque immeuble i est représenté par un triplet (h_i, g_i, d_i) où h_i représente la hauteur, g_i l'abscisse gauche et d_i l'abscisse droite. On cherche à écrire un algorithme SILHOUETTE(l) qui, étant donnée une liste de tels triplets, renvoie une liste de paires (x_j, y_j), triés par x_j croissants, et correspondant aux changements de niveau de la silhouette de gauche à droite.

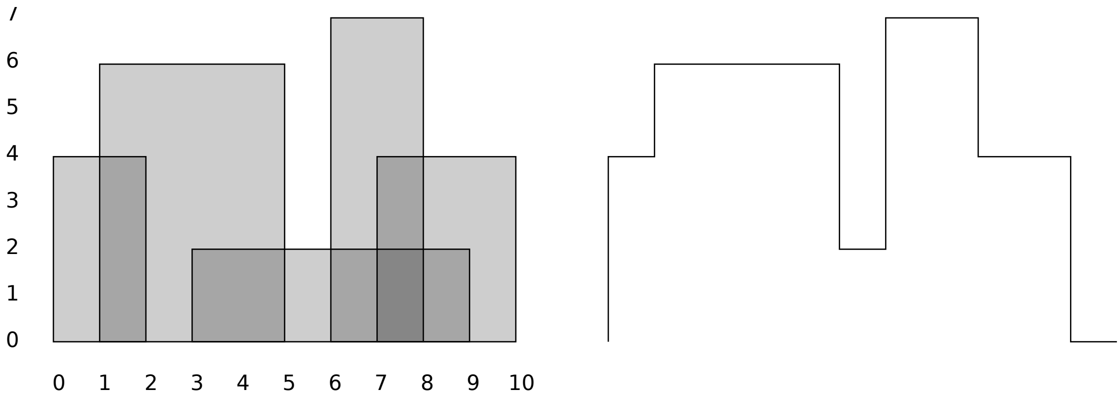


FIGURE 1 – Un ensemble d'immeubles (à gauche) et la silhouette correspondante (à droite).

QUESTION 5 – Donner la liste de triplets représentant l'ensemble d'immeubles de la figure 1, ainsi que la liste de paires représentant la silhouette.

QUESTION 6 – Étant données deux silhouettes s_1 et s_2 représentées par une liste de paires, écrire une fonction FUSION-SILHOUETTE(s_1, s_2) qui renvoit une nouvelle liste de paires s, représentant la fusion des deux silhouettes.

QUESTION 7 – En déduire l'écriture de la fonction SILHOUETTE(l), et calculer la complexité associée.

Problèmes ouverts (Bonus)

5 Vote majoritaire

Le but du problème est de trouver dans un tableau T de taille $n = |T|$ l'élément majoritaire, c'est-à-dire apparaissant strictement plus de $\lfloor n/2 \rfloor$ fois, s'il en existe un.

QUESTION 8 – Proposer une solution naive (sans diviser pour régner). Quelle en est la complexité ?

QUESTION 9 – Proposer une solution utilisant une approche « diviser pour régner » de complexité asymptotique optimale.

QUESTION 10 – Proposer un algorithme de complexité asymptotique optimale, qui fonctionne sans modifier le tableau et à coût mémoire constant.

6 Algorithme de Strassen

Dans cet exercice, on s'intéresse au problème de multiplication de deux matrices carrées X et Y de taille $n \times n$.

QUESTION 11 – Quel est la complexité de l'algorithme naïf ?

QUESTION 12 – Proposer un algorithme diviser pour régner en $\mathcal{O}(n^3)$.

QUESTION 13 – Proposer un algorithme diviser pour régner plus astucieux et donner sa complexité.

Indice :

$$XY = \begin{pmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{pmatrix}$$

avec :

— $X = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$

— $Y = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$

— $P_1 = A(F - H)$
— $P_2 = (A + B)H$
— $P_3 = (C + D)E$
— $P_4 = D(G - E)$
— $P_5 = (A + D)(E + H)$
— $P_6 = (B - D)(G + H)$
— $P_7 = (A - C)(E + F)$