Aix Marseille université	Domaine Sciences et Technologies LICENCE 3 INFORMATIQUE **Algorithmique 2 : TD 6** **Code UE : SIN5U03C**
Année 2020-2021	**Diviser pour régner, enveloppes convexes**

Exercice 1 (Relations de récurrence)

Trouver les complexités asymptotiques induites par les relations de récurrence suivantes.

$$T(n) = 2T\left(\frac{n}{2}\right) + n^4 \qquad T(n) = T\left(\frac{7n}{10}\right) + n$$

$$T(n) = 16T\left(\frac{n}{4}\right) + n^2 \qquad T(n) = 7T\left(\frac{n}{3}\right) + n^2$$

$$T(n) = 2T\left(\frac{n}{4}\right) + 1 \qquad T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$$

$$T(n) = 2T\left(\frac{n}{4}\right) + n \qquad T(n) = 2T\left(\frac{n}{4}\right) + n^2$$

Exercice 2 (Tri des complexités) Trier les fonctions $n^3 + \sqrt{n}, n^2 + \log n, n \log n, 3^n, \sqrt{n} + \log n, 4^n$ suivant leur ordre de grandeur asymptotique : f sera *avant* g si $f \in O(g)$.

Exercice 3 (Nombre d'inversions dans un tableau d'entiers)

Une inversion dans un tableau d'entiers T indicé de 0 à $n-1$ est une paire d'indices $i, j \in [0, n-1]$ telle que $i < j$ et $T[i] > T[j]$. On se propose de calculer le nombre d'inversions d'un tableau par différentes méthodes.

(a) Quel est le nombre d'inversions dans $[1, 5, 4, 3, 6, 2]$?

(b) Proposer un algorithme non-récursif et donner sa complexité.

(c) Proposer un algorithme récursif tel qu'un appel sur un tableau de taille n fait au plus un appel récursif, sur un tableau de taille $n-1$. Analyser sa complexité.

(d) Proposer un algorithme récursif tel que chaque appel sur un tableau de taille n fait au plus deux appels récursifs, sur des tableaux de taille $\frac{n}{2}$. À nouveau, analyser sa complexité.

Exercice 4 Un tableau à n elements contient des 1 suivis par des 0 (par example, $\{1, 1, 1, 1, 1, 0, 0, 0, 0, 0\}$). Calculer le nombre des 0 dans ce tableau en $O(\log n)$ temps.

Exercice 5 (Sous-tableau maximum) Soit T un tableau de n nombres positifs ou négatifs (indicés de 1 à n). Nous cherchons à trouver un sous-tableau qui maximise la somme de ses éléments. De manière plus précise, nous cherchons 2 indices $i, j \in \{1, \ldots, n\}$, avec $i \leq j$, et tels que $s_{i,j}$ soit maximal :

$$s_{i,j} = T_i + T_{i+1} + \cdots + T_j = \sum_{k=i}^{j} T_k$$

Par exemple si $T = [-2, -4, 3, -1, 5, 6, -7, -2, 4, -3, 2]$ alors le résultat est $(3, 6)$, car $s_{3,6} = [3, -1, 5, 6]$ est de somme 13, ce qui est le maximum possible. *Dans tout ce qui suit on s'intéressera à la complexité en nombre d'additions*

Une méthode naïve Décrire une méthode naïve (donc simple) pour résoudre ce problème. Quelle est sa complexité asymptotique (en nombre d'additions) en fonction de n ?

Diviser pour régner Un première idée pour améliorer la méthode naïve consiste à découper le tableau en deux et à chercher la plus grande somme sur chacun des sous-tableaux. Nous pouvons le décrire ainsi :

$$\begin{aligned}\texttt{SousTableauMaximum}(T, 1..n) = \max\{&\texttt{SousTableauMaximum}(T, 1..n/2-1),\\ &\texttt{SousTableauMaximum}(T, n/2+1..n),\\ &\texttt{SousTableauMaximumContenantLaCaseDuMilieu}(T, 1..n)\}\end{aligned}$$

La fonction `SousTableauMaximumContenantLaCaseDuMilieu` doit trouver le sous-tableau (i, j) maximisant $s_{i,j}$, et tel que $i \leq n/2 \leq j$.

(a) Écrire le pseudo code réalisant `SousTableauMaximumContenantLaCaseDuMilieu`. Quel est sa complexité asymptotique en nombre d'additions ?

(b) Donner l'équation de récurrence permettant de trouver la complexité totale de la recherche du sous-tableau maximum avec cette méthode. Cela ressemble à la complexité d'un algoritme connu, lequel ? Résoudre cette équation pour trouver une formule asymptotique de la complexité de cet algorithme.

Programmation dynamique On va essayer de faire mieux en utilisant la programmation dynamique. Pour cela considérons :

$$M_i = \max_{j \geq i} s_{i,j} = \max\{s_{i,i}, s_{i,i+1}, \ldots, s_{i,n}\}$$

M_i représente donc la somme maximale que l'on peut atteindre pour un sous-tableau commençant à l'indice i.

(a) Supposons que nous avons calculé le tableau $M = [M_1, M_2, \ldots, M_n]$. Comment résoudre le problème du sous-tableau maximum de T en utilisant M ?

(b) Exprimer la valeur de M_i en fonction de $M_{i+1}, M_{i+2}, \ldots, M_n$ et de la valeur T_i.

(c) En déduire un pseudocode permettant de calculer M_i pour tout i. Quelle est sa complexité ?

(d) Écrire l'algorithme complet permettant de calculer les indices (i, j) du sous-tableau maximum d'un tableau T, et donner sa complexité.

(e) Que se passe-t-il si toutes les valeurs du tableau T sont négatives ?

Exercice 6 (L'élément médian de deux tableaux triés) Etant données deux tableaux triés $A[n]$ et $B[n]$ de taille n chacun, trouver l'élément médian du tableau de taille $2n$ résultant de la fusion des tableaux A et B.

(i) Proposer d'abord un algorithme en $O(n)$.

(ii) En utilisant la méthode *diviser-pour-regner* proposer et justifier un algorithme en $O(\log n)$.

Rappel : L'élément *médian* d'un tableau $C[m]$ de taille m est l'élément de rang $\lfloor m/2 \rfloor$ du tableau trié à partir de C.

Exercice 7 Un tableau T à n éléments contient les éléments d'une suite arithmétique avec un élément manquant. En utilisant la méthode *diviser-pour-regner* proposer un algorithme en $O(\log n)$ pour trouver cet élément. Par example, si $T = \{2, 4, 6, 10, 12, 14, 16\}$, alors l'algorithme doit retourner 8.

Exercice 8 (Enveloppes convexes en $\mathbb{R}^2$) L'*enveloppe convexe* $\mathrm{conv}(P)$ d'un ensemble de points $P = \{p_1, p_2, \ldots, p_n\}$ de $\mathbb{R}^2$ est le plus petit polygone convexe qui les contient tous. Les sommets de $\mathrm{conv}(P)$ sont les *points extrémaux* de P.

(a) Quelle est la structure des données appropriée pour représenter $\mathrm{conv}(P)$?

(b) Comment trouver rapidement (en $O(n)$) un point extrémal de P ?

(c) Comment, en partant d'un point extrémal, construire l'intégralité de $\mathrm{conv}(P)$? Quelle est la complexité de l'algorithme ?

Exercice 9 (Enveloppes convexes en $\mathbb{R}^2$ par la méthode incrementale) Supposons qu'on souhaite construire $\mathrm{conv}(P)$ de façon incrémentale. Pour cela on trie les points de P par abscisse croissante. Supposons, sans perte de généralité, que $p_1, p_2, \ldots, p_n$ est la liste triée des points de P. Soit $Q_i = \mathrm{conv}(p_1, \ldots, p_i)$.

(a) Comment construire Q_{i+1} à partir de Q_i et p_{i+1} ?

(b) Quelle est la complexité de construction de $\mathrm{conv}(P) = Q_n$ en utilisant cette méthode ?

Exercice 10 (Enveloppes convexes en $\mathbb{R}^2$ par diviser-pour-régner) Supposons maintenant qu'on souhaite construire $\mathrm{conv}(P)$ par la méthode diviser-pour-régner. Pour cela, de nouveau, on trie les points de P par abscisse croissante.

(a) Comment utiliser le tri pour pouvoir séparer P en deux ensembles P_1 et P_2 de même taille en utilisant une droite verticale (c'est-à-dire par abscisse) ?

(b) Etant donnés $Q_1 = \mathrm{conv}(P_1)$ et $Q_2 = \mathrm{conv}(P_2)$, comment construire $\mathrm{conv}(P) = \mathrm{conv}(Q_1 \cup Q_2)$?

(b) Décrire en pseudocode un algorithme diviser-pour-régner pour construire $\mathrm{conv}(P)$. Quelle est sa complexité ?