NOM : PECATTE **Prénom :** Timothée **Jury :**

Algèbre ← Entourez l'épreuve → **Analyse**

Sujet choisi : 902. Diviser pour régner.

Autre sujet :

I. Introduction et outils :

I.1. Motivations :

Ex 1 : Dichotomie pour trouver un élément dans une liste triée

Ex 2 : Trier une liste d'éléments en triant d'abord des sous listes.

Paradigme 3 : Diviser pour régner s'effectue en 3 étapes :
- Diviser en sous-problèmes indépendants
- Résoudre les sous-problèmes
- Recombiner les solutions des sous-problèmes en une solution au problème initial.

Rem 4 : On aboutit à des complexités satisfaisant des relations de récurrence de la forme : $T(n) = f(n) + \sum_{i=1}^{\ell} a_i T(\frac{n}{b_i})$

I.2. Outils mathématiques :

Méthode 5 : Substitution
↳ "Conjecturer la forme de la solution et injecter"

Ex 6 : $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + n$ en devinant $T(n) \leq cn \lg n$

Méthode 7 : Changement de variable
↳ "Se ramener à une récurrence plus simple"

Ex 8 : $T(n) = 2T(\lfloor \sqrt{n} \rfloor) + \lg n$, on pose $n = 2^m$ et $S(m) = T(2^m)$ pour avoir $S(m) = 2S(m/2) + m$

Méthode 9 : Arbre récursif
↳ "Dépliage de tous les appels récursifs"

Méthode / Th 10 : Méthode générale si $T(n) = aT(n/b) + f(n)$
- Si $f(n) = O(n^{\log_b a - \varepsilon})$ pour $\varepsilon > 0$, alors $T(n) = \Theta(n^{\log_b a})$
- Si $f(n) = \Theta(n^{\log_b a})$, alors $T(n) = \Theta(n^{\log_b a} \lg n)$
- Si $f(n) = \Omega(n^{\log_b a + \varepsilon})$ pour $\varepsilon > 0$ et si $af(n/b) \leq cf(n)$ pour $c < 1$, alors $T(n) = \Theta(f(n))$

Rem 11 : $n^{\log_b a}$ est une valeur critique

Ex 12 : $T(n) = 2T(n/2) + n\lg n$ on ne peut pas utiliser le Th 10.

II. Applications à des séries de données :

Pb 13 : (Tri) Étant donné l une liste d'entiers de taille n, renvoyer la liste triée

Algo 14 : TRI-FUSION résoud le pb 13 en temps $\Theta(n \lg n)$

Rem 15 : Complexité optimale mais trop lent en pratique

Pb 16 : Étant donné une liste d'entiers, renvoyer la valeur de l'élément maximal et de l'élément minimal

Algo 17 : Le pb 16 peut être résolu avec le paradigme D&R avec $\frac{3n}{2} - 2$ comparaisons

Rem 18 : Complexité optimale

DEV 1

Pb 19 : Étant donné une liste d'éléments, trouver la médiane

Algo 20 : Le pb 19 peut être résolu en temps $\Theta(n)$

Pb 21 : Trouver le $k^{\text{ème}}$ plus petit élément d'une liste d'entiers.

Algo 22 : Le pb 21 peut être résolu en suivant le paradigme D&R en $O(n)$.

III. Applications au calcul formel :

III.1. Exemples classiques :

Pb 23 : Calculer a^b, avec $a, b \in \mathbb{N}$

Algo 24 : EXPO-RAPIDE résout le pb 23 en $O(\log b)$ multiplications.

Rem 25 : Cet algo est plus stable car il multiplie des nombres de même taille

Pb 26 : Calculer AB, où $A, B \in \mathcal{M}_n(\mathbb{K})$

Algo 27 : STRASSEN résout le pb 26 en temps $\Theta(n^{\log 7}) = \Theta(n^{2,81})$ via la récurrence $T(n) = 7T(n/2) + \Theta(n^2)$

III.2. Polynômes :

Pb 28 : Calculer $P.Q$, avec $P, Q \in \mathbb{K}_n[X]$, qui sont donnés par la liste de leurs coefficients

Algo 29 : Karatsuba résout le pb 28 en temps $\Theta(n^{\log 3}) = \Theta(n^{1,58})$ via la récurrence $T(n) = 3T(n/2) + \Theta(n)$

DEV 2

Algo 30 : FFT résout pb 28 en temps $\Theta(n \log n)$ si $\mathbb{K} = \mathbb{R}$

Pb 31 : Évaluer $P \in \mathbb{K}_n[X]$ en $x_1, \dots, x_n \in \mathbb{K}$

Algo 32 : Le pb 31 peut être résolu via D&R en temps $O(M(n) \log n)$ où $M(n)$ est le coût de la multiplication de deux polynômes de degré n.

IV. Applications à la géométrie :

Pb 33 : Étant donné n polygones convexes à k sommets, calculer leur intersection

Algo 34 : Le pb 33 peut être résolu via D&R en temps $O(kn \log n)$ via la récurrence $\begin{cases} T(n) = 2T(n/2) + \Theta(kn) \\ T(2) = \Theta(k) \end{cases}$

Rem 35 : C'est optimal car on peut réduire Pb 13 à Pb 33.

Pb 36 : (Skyline) Étant donné n rectangles, dont les bases sont sur l'axe des abscisses, déterminer l'enveloppe des rectangles

Algo 37 : Le pb 36 se ramène à une fusion de polygones, qui peut être résolu via D&R en temps $\Theta(n \log n)$ via la récurrence $T(n) = 2T(n/2) + \Theta(n)$.

Rem 38 : Optimal car là aussi Pb 13 se réduit à pb 36.

Pb 39 : Étant donné un ensemble de n points, déterminer l'enveloppe convexe de ces points.

Algo 40 : Le pb 39 peut être résolu en temps $\Theta(n)$ via D&R.

Rem 41: La solution donne même $\Theta(n_1+n_2)$ où n_1 et n_2 sont le nombre de points sur l'enveloppe convexe des deux sous-polygones considérés.

Pb 42: Étant donné n points dans le plan, trouver les deux points les plus proches

Algo 43: Le pb 42 peut être résolu en temps $\Theta(n \log n)$ via D&R.

Pb 44: Étant donné n points dans le plan, calculer leur diagramme de Voronoï

Algo 45: Le pb 44 se résout en temps $\Theta(n \log n)$ via D&R

Rem 46: C'est optimal car le pb 43 se réduit au pb 42, qui se réduit au pb 44.

V. Comparaisons aux autres paradigmes :

V.1. Méthode gloutonne :

- Approche "bas vers haut" contre "haut vers bas" dans D&R
- Récurrence de la forme $T(n) = T(n-1) + f(n)$
- Sous-structure optimale
- Algorithmes plus simples à mettre en place mais correction plus dure

V.2. Programmation dynamique :

- Sous-problèmes non-indépendants.
- Plusieurs paramètres pour la complexité.

Réfs :

- Cormen
- Algorithms from P to NP, Moret.

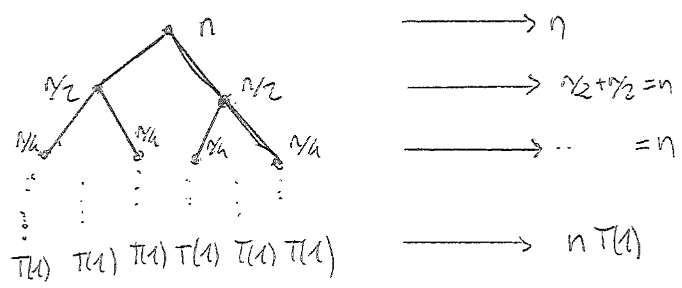
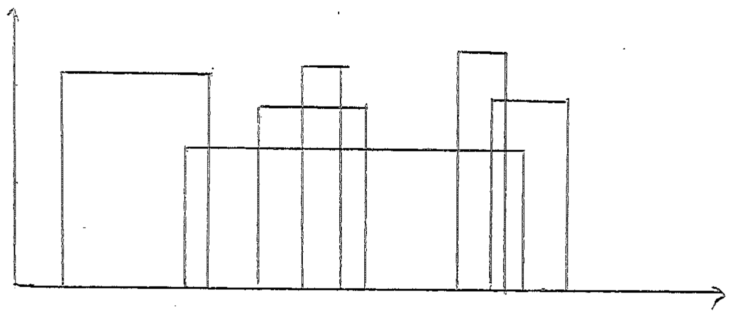


Figure 1: Représentation de la méthode 9 pour $T(n) = 2T(n/2) + n$.



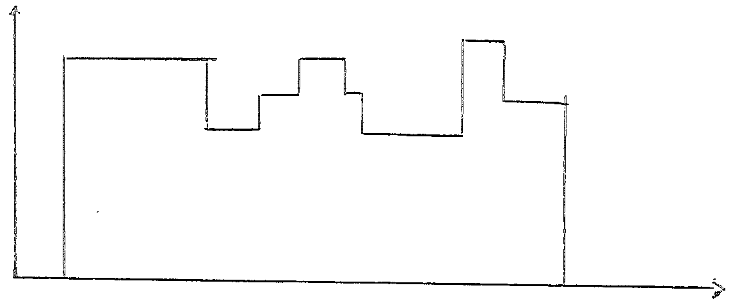


Figure 2: Représentation du pb 36

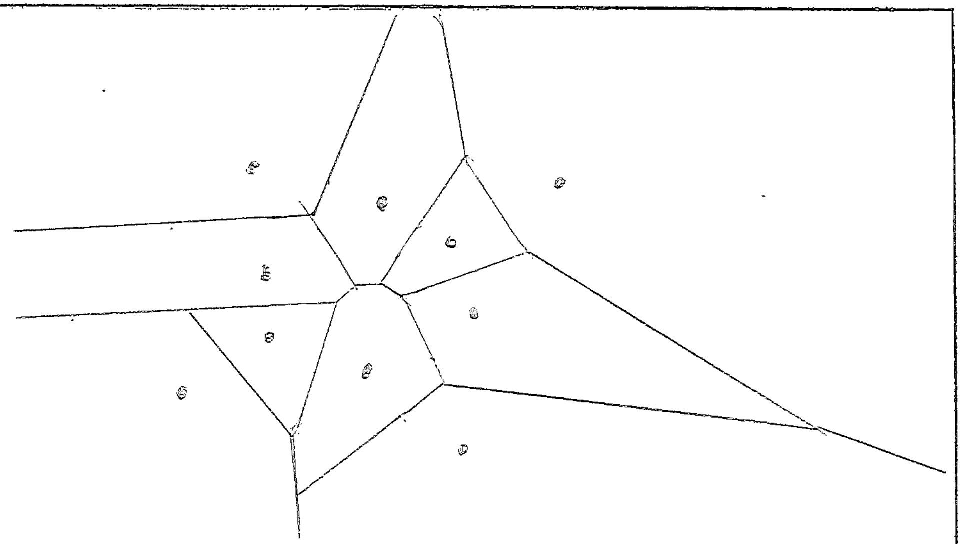


Figure 3 : Diagramme de Voronoï, pb 44