I] Induction et récurrence

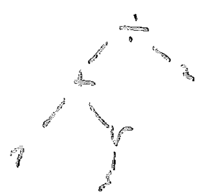
Def Un algo. « Diviser pour régner » consiste à résoudre un problème en : 1) le découpant en sous parties (Diviser)
2) résoudre récursivement les sous parties (Régner)
3) fusionner les résultats.

Algo DR(P)

- $[P_1, \dots, P_N] \leftarrow$ Diviser(P)
- $[R_1, \dots, R_N] \leftarrow$ Map DR $[P_1, \dots, P_N]$
- $R \leftarrow$ Fusionner $[R_1, \dots R_N]$

⚠ ne pas oublier le cas d'arrêt

Ex Arbre syntaxique (cf Euclide) : Tracer $\frac{1+\sqrt{5}}{2}$
Θ(Noeuds)



Ex Récursion $n! = n.(n-1)!$ $\qquad$ max(t::l) = max(t, max l)
$\Theta(N)$

Ex La recherche dichotomique : T trié et $D = D_1 D_2$ avec $|D_1| = |D_2|$
$\Theta(\cancel{N} \log N)$ (cas trivial de DPR)

Def (informelle) : Un algo. « Diviser pour régner » est un algo. inductif sur une structure inductive non canonique.

II] Complexité et algorithme de tri

Algo : Tri fusion

↳ un tableau est constitué de deux demi tableaux

↳ si $|T| = 2^p$ alors $\Theta(2^p \cdot p) = \Theta(N \log N)$

avec $|T| = N$

Rq : Même si $|T|$ n'est pas une puissance de 2 on a toujours $\Theta(N \lg N)$

Cas général . Soit P un problème de taille N vérifiant :

- Division P est en $\Theta(N)$
- Fusion$(R_1, \dots, R_N)$ est en $\Theta(N)$
- $\sum |P_i| \leq |P|$
- $\forall i, \quad P_i \leq \lceil \frac{P}{d} \rceil$ avec $d \in \mathbb{N}^* \setminus \{1\}$
  (d ne dépend pas de P)

Alors on a $\Theta(N \lg N)$

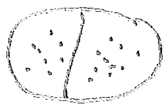
Algo : Tri rapide
↳ l'opération Division modifie le tableau : $\boxed{A \,|\, p \,|\, B}$ tq $\forall x,y \in A \times B, \; x \leq p \leq y$
↳ $\Theta(n^2)$
↳ Cependant en moyenne on a du $\Theta(n \lg n)$

III) Application géométrique

A1 **Problème** : Soit $\mathcal{P}$ un ensemble de points du plan :
trouver $\underset{p,q \in \mathcal{P}}{\mathrm{Argmin}}\ dist(p,q)$ = trouver p et q, $dist(p,q) = \min$

Problème : Soit $\mathcal{P}$ un ensemble de points du plan. Trouver l'enveloppe convexe.

Dans les deux cas si $|\mathcal{P}| = n$ on a $\Theta(n \lg n)$ et $\mathcal{P} = \mathcal{P}_{gauche} \uplus \mathcal{P}_{droite}$
avec $|\mathcal{P}_{gauche}| = |\mathcal{P}_{droite}|$



IV) Application algébrique

Problème Exponentiation rapide : $\begin{cases} X^{2n+1} = X \cdot X^n \cdot X^n \\ X^{2n} = X^n \cdot X^n \\ X^0 = 1 \end{cases}$

$\Theta(\log n)$

Δ2 **Algo** : Transformée de Fourier rapide

Soit $A = [a_0 \dots a_{N-1}]$ un vecteur. On veut calculer pour tout k $\sum_{j=0}^{n-1} a_j \left(e^{\frac{2i\pi}{n} \cdot k}\right)^j = y_k = y[a]_k$

↳ En posant $B = [b_0 \dots b_{\frac{N}{2}-1}] = [a_0, a_2 \dots a_{N-2}]$
et $C = [c_0 \dots c_{\frac{N}{2}-1}] = [a_1, a_3, \dots, a_{N-1}]$

on a
$$\begin{cases} \underbrace{y[a]_k}_{N} = \underbrace{y[b]_k}_{N/2} + e^{\frac{2ik\pi}{n}} \underbrace{y[c]_k}_{N/2} & \text{pour } \underline{k \leq \frac{N}{2}-1} \\ y[a]_{k+\frac{N}{2}} = y[b]_k - e^{\frac{2ik\pi}{n}} y[c]_k & \text{pour } k \leq \frac{N}{2}-1 \end{cases}$$

↳ $\Theta(n \log n)$

Δ3 ## V) Calcul de l'automate minimal

Problème : Soit $\mathcal{A} = (P, Q, \Sigma, I, F, \cancel{\ })$ un automate émondé-déterministe. On pose $p \sim q$ si $L_p = L_q$ avec L_p (resp L_q) le langage reconnu partant de p (resp q).

Def : Automate minimal : (P', Q', Σ, I', F') avec $P' = P/\sim$ $I' = I/\sim$ $F' = F/\sim$ et Q' défini par $\bar{p} \xrightarrow{a} \bar{q}$ ssi $p \xrightarrow{a} q$

Bien défini.

Algo calcul de l'automate minimal en $n \log(n)$ ← nb d'états
↳ (de Hopcroft) $O(|\Sigma| n \log n)$

Refs. Intro à l'algo de Cormen (Δ1 et 2)
Éléments d'algo de Beauquier (Δ3)