Chapitre 4

Relations de récurrence

Synthèse: Math

Jerome Jadot
[Date]

Table des matières

Position générale

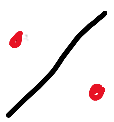
Droites en position générale

Définition

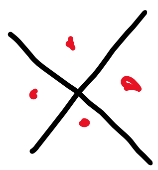
"Un ensemble de droites du plan est en position générale si toute paire de droites s'intersectent en exactement un point et tout triple de droites ont une intersection vide."

$\Phi2(n)$ = nombre de régions du plan délimitées par n droites en position générale.

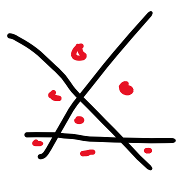
Exemple



$\Phi2(1) = 2$



$\Phi2(2) = 4$



$\Phi2(3) = 7$

n	0	1	2	3
$\Phi2(n)$	1	2	4	7

On remarque que la différence entre Φ2(n) et Φ2(n-1) est n.

Théorème 1

Théorème 1

Pour tout $n \in \mathbb{N}$, le nombre de régions du plan délimitées par n droites en position générale ne dépend pas du choix de ces droites. En d'autres termes, $\Phi_2(n)$ est bien défini. De plus,

$$\Phi_2(n) = \Phi_2(n-1) + n$$

pour $n \geqslant 1$, et $\Phi_2(0) = 1$.

Pour résoudre la récurrence obtenue ci-dessus pour $\Phi_2(n)$, on la déroule :

$$\begin{aligned}
\Phi_2(n) &= n + \Phi_2(n-1) \\
&= n + (n-1) + \Phi_2(n-2) \\
&\vdots \\
&= n + (n-1) + (n-2) + \cdots + 1 + \overbrace{\Phi_2(0)}^{=1} \\
&= \frac{n(n+1)}{2} + 1 \\
&= \frac{n(n-1)}{2} + n + 1 \\
&= \binom{n}{2} + \binom{n}{1} + \binom{n}{0} .
\end{aligned}$$

Plan en position générale

Définition

"Un ensemble de plans de l'espace est en position générale si toute paire de plans s'intersectent en une droite, tout triple de plans s'intersectent en un point et tout quadruple de plans ont une intersection vide."

Φ3(n) = nombre de régions de l'espace délimitées par n plans en position générale.

Φ3(n) = Φ3(n-1) + Φ2(n-1)   ∀n >= 1; Φ3(0) = 1

En déroulant, on obtient une formule pour $\Phi_3(n)$:

$$\begin{aligned}\Phi_3(n) &= \Phi_2(n-1) + \Phi_3(n-1)\\ &= \binom{n-1}{2} + \binom{n-1}{1} + \binom{n-1}{0} + \Phi_3(n-1)\\ &= \left[\binom{n-1}{2} + \binom{n-1}{1} + \binom{n-1}{0}\right] +\\ &\quad \left[\binom{n-2}{2} + \binom{n-2}{1} + \binom{n-2}{0}\right] + \ldots +\\ &\quad \underbrace{\left[\binom{0}{2} + \binom{0}{1} + \binom{0}{0}\right]}_{\Phi_2(0)} + \underbrace{\Phi_3(0)}_{=1=\binom{n}{0}}\\ &= \binom{n}{3} + \binom{n}{2} + \binom{n}{1} + \binom{n}{0}.\end{aligned}$$

Généralisation en dimension d

Tout cela se généralise en dimension d pour la récurrence :

$\Phi d(n) = \Phi d(n-1) + \Phi d-1(n-1)$ $\forall n >= 1$; $\Phi d(0) = 1$

Et pour la formule :

$$\Phi_d(n) = \binom{n}{d} + \binom{n}{d-1} + \ldots + \binom{n}{1} + \binom{n}{0}.$$

Fibonacci

Définition

La suite des nombres de Fibonacci est l'unique suite (Fn) n∈N solution pour la récurrence :

$F_n = F_{n-1} + F_{n-2}$ $\forall n >= 2$; $F_0 = 0$; $F_1 = 1$

Pour Fn/Fn-1, plus n est grand plus le quotient se rapproche du nombre d'or.

Le Nombre d'Or :

$$\varphi = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \ddots}}}$$

$$\varphi = 1 + \cfrac{1}{\boxed{1 + \cfrac{1}{1 + \cfrac{1}{1 + \ddots}}}}$$

Le Nombre d'Or : Résoudre l'équation suivante pour φ.

$$\varphi = 1 + \frac{1}{\varphi}.$$

$\Rightarrow \varphi^2 = \varphi + 1$
$\Rightarrow \varphi^2 - \varphi - 1 = 0$
$\Delta = B^2 - 4AC$
$= 1 - 4.1.(-1) = 5$

$$\boxed{\varphi_{1,2} = \frac{1 \pm \sqrt{\Delta}}{2}}$$

Formule explicite pour le n-ème nombre de Fibonacci Fn

On cherche des solutions de la forme $F_n = \beta^n$ pour certains $\beta \in \mathbb{R} \setminus \{0\}$. Réécrivons l'équation $F_n = F_{n-1} + F_{n-2}$ en remplaçant F_n par β^n :

$$\begin{array}{rrcl} & \beta^n & = & \beta^{n-1} + \beta^{n-2} \quad \forall n \geqslant 2 \\ \overset{\beta\neq 0}{\Longleftrightarrow} & \beta^2 & = & \beta + 1 \\ \Longleftrightarrow & \beta & = & \frac{1 \pm \sqrt{5}}{2} . \end{array}$$

Définition : le nombre d'or

Posons $\varphi := \frac{1+\sqrt{5}}{2}$ et $\bar{\varphi} := \frac{1-\sqrt{5}}{2}$. L'irrationnel φ est le ***nombre d'or****. L'irrationnel $\bar{\varphi}$ est le* ***conjugué*** *du nombre d'or φ.*

Théorème

Théorème 4

Pour tout $n \in \mathbb{N}$,

$$F_n = \frac{1}{\sqrt{5}}[\varphi^n - \bar{\varphi}^n] = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right] .$$

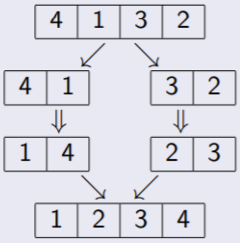
Tri-fusion

C'est un algorithme de tri qui pour trier N, trie N/2 et N/2.

Exemple

Exemple 5

Voyons un exemple pour $N = 4$.



Pour le moment, on prend N = 2^n (n>=0). Etant donné la structure récursive de l'algo nous pouvons écrire : $C_N = C_{\lceil N/2 \rceil} + C_{\lfloor N/2 \rfloor} + N \quad \forall N \geqslant 2; \quad C_1 = 0$.

Pour N = 2^n, les plancher et plafond disparaissent :

$$C_{2^n} = 2C_{2^{n-1}} + 2^n \quad \forall n \geqslant 1; \quad C_{2^0} = 0$$

Après division par 2^n :

$$\frac{C_{2^n}}{2^n} = \frac{C_{2^{n-1}}}{2^{n-1}} + 1 \quad \forall n \geqslant 1; \quad \frac{C_{2^0}}{2^0} = 0$$

Après réflexion on peut poser $a_n := \frac{C_{2^n}}{2^n}$.

Donc : $a_n = 1 + a_{n-1}$

$= 1 + 1 + a_{n-2}$

$= 1 + 1 + 1 + \dots + 1 + a_0$ ($a_0 = 0$)

$= n$

En revenant aux variables d'origine, on a $C_{2^n} = 2^n a_n = 2^n n$.

Etant donné que N = 2^n, on a n = $\log_2 N$ et on obtient alors :

Théorème

"Le nombre de copies effectuées par le tri-fusion sur un vecteur de taille N = 2^n est : $C_n = N\log_2 N$"

Récurrences linéaires du premier ordre

Théorème

Epsilon = symbole de sommation et Pi = symbole de multiplication

Théorème 7

La récurrence

$$x_n = c_n x_{n-1} + d_n \quad \forall n \geqslant 1; \quad x_0 = 0$$

a pour solution explicite

$$x_n = \sum_{i=1}^{n} d_i \prod_{j=i+1}^{n} c_j = d_n + d_{n-1}c_n + d_{n-2}c_{n-1}c_n + \cdots + d_1 c_2 \cdots c_n.$$

Remarque : Si $x_0 \neq 0$, alors on décale les indices des x_n, c_n, d_n jusqu'à obtenir $x_0 = 0$.

Récurrences linéaires homogènes à coefficients constants

En général, une RLCC homogène s'écrit :

$$x_n = c_{d-1}x_{n-1} + c_{d-2}x_{n-2} + \cdots + c_0 x_{n-d} \quad \forall n \geqslant d \ ,$$

Ou encore

$$-x_n + c_{d-1}x_{n-1} + c_{d-2}x_{n-2} + \cdots + c_0 x_{n-d} = 0 \quad \forall n \geqslant d \ . \quad (2)$$

Les nombres $c_0, c_1, \ldots, c_d$ sont des constantes. $c_0, c_1, \ldots, c_d \in$ C (C nombre complexe) (en pratique les coefficients seront entiers). Si c_0 est différent de 0, l'ordre d'une telle récurrence est d.

Définition : Polynôme caractéristique

Définition 4 (Polynôme caractéristique)

Le ***polynôme caractéristique*** *de la RLCC (2) est*

$$p(t) := -t^d + c_{d-1}t^{d-1} + c_{d-2}t^{d-2} + \cdots + c_0 = \sum_{i=0}^{d} c_i t^i \ ,$$

où $c_d := -1$.

Théorème

Théorème 8

Notons S l'ensemble des solutions de la RLCC (2). Alors :

(i) *S est un sous-espace vectoriel de $\mathbb{C}^{\mathbb{N}}$,*

(ii) $\dim(S) = d$.

Définition : multiplicité

Définition 5 (Multiplicité)

Pour rappel, un nombre complexe $a \in \mathbb{C}$ est ***une racine de multiplicité*** *$m = m(a)$ d'un polynôme complexe $p(t) \in \mathbb{C}[t]$ si $p(t)$ est divisible par $(t-a)^m$, mais pas par $(t-a)^{m+1}$.*

Les lemma

Lemma 2

Soit $p(t) \in \mathbb{C}[t]$ un polynôme complexe de degré $d \geqslant 1$. Un nombre complexe a est une racine de multiplicité $m = m(a)$ du polynôme $p(t)$ si et seulement si $p(a) = \frac{\mathrm{d}p}{\mathrm{d}t}(a) = \cdots = \frac{\mathrm{d}^{m-1}p}{\mathrm{d}t^{m-1}}(a) = 0$ et $\frac{\mathrm{d}^m p}{\mathrm{d}t^m}(a) \neq 0$.

Lemma 3

Pour tout entier $j \geqslant 1$, il existe des coefficients entiers $\alpha_1, \alpha_2, \ldots, \alpha_j$ tels que

$$\begin{aligned} i^j &= \alpha_1\, i + \alpha_2\, i(i-1) + \cdots + \alpha_j\, i(i-1)\cdots(i-j+1) \\ &= \sum_{k=1}^{j} \alpha_k i(i-1)\cdots(i-k+1)\,. \end{aligned}$$

Lemma 4

Si $a \in \mathbb{C}$ est une racine de multiplicité $m \geqslant 1$ du polynôme $\sum_{i=0}^{d} c_i t^i \in \mathbb{C}[t]$ de degré d, alors

$$\sum_{i=0}^{d} c_i i^j a^i = 0$$

pour $j \in \{0, \ldots, m-1\}$ et

$$\sum_{i=0}^{d} c_i i^m a^i \neq 0\,.$$

Théorème

Théorème 9 (Récurrences linéaires homogènes à coefficients constants)

Considérons la RLCC homogène d'ordre $d \geqslant 1$:

$$-x_n + c_{d-1}x_{n-1} + c_{d-2}x_{n-2} + \cdots + c_0 x_{n-d} = 0 \quad \forall n \geqslant d \ .$$

où $c_0, \ldots, c_{d-1} \in \mathbb{C}$ et $c_0 \neq 0$. Notons $p(t) := \sum_{i=0}^{d} c_i t^i \in \mathbb{C}[t]$ son polynôme caractéristique, où $c_d := -1$. Toute solution de cette relation de récurrence est une combinaison linéaire des d suites de la forme $(n^j \beta^n)_{n \in \mathbb{N}}$, où β est une racine de $p(t)$ et $j \in \{0, 1, \ldots, m(\beta) - 1\}$, c.-à-d. j est un naturel strictement inférieur à la multiplicité de β.

Solution générale d'une RLCC homogène :

$$x_n = \sum_{\beta \text{ racine}} \sum_{j=0}^{m(\beta)-1} \lambda_{\beta,j}\, n^j \beta^n$$

$\lambda_{\beta,j}$ sont des coefficients complexes déterminés dès que l'on possède d conditions initiales x_0, …, x_d .

Exemple

Exemple 10

$x_n + x_{n-1} + x_{n-2} = 0$ pour $n \geqslant 2$ avec $x_0 = 0$ et $x_1 = 1$. La solution est

$$x_n = \frac{i}{\sqrt{3}} \left(\left(\frac{-1-\sqrt{3}i}{2} \right)^n - \left(\frac{-1+\sqrt{3}i}{2} \right)^n \right) \ .$$

Horreur ? Non, la suite est toujours à valeurs entières. Si on s'interdisait d'utiliser les complexes, on se priverait d'une belle formule !

Exemple 11

Quelle est la solution de $x_n = 5x_{n-1} - 6x_{n-2}$ pour $n \geqslant 2$ avec $x_0 = 0$ et $x_1 = 1$?

$$x_n = 3^n - 2^n \ .$$

Récurrence diviser-pour-régner

https://www.youtube.com/watch?v=u_U3NJn2Zu8&ab_channel=EventsBeats

Pour localiser un nombre x dans un vecteur trié de taille N, l'algo de recherche binaire compare x au $\lceil N/2 \rceil$-ème élément du vecteur (appelé pivot).

- Si x = pivot, c'est bon l'algo a trouvé x
- Si x < pivot, l'algo s'appelle récursivement sur le sous-vecteur constitué des éléments en postions 1 jusque $\lceil N/2 \rceil$-1

- Si x > pivot, l'algo s'appelle récursivement sur le sous-vecteur constitué des éléments en postions 1 jusque $\lceil N/2 \rceil$+1

Théorème

Théorème 12

Le nombre de comparaisons effectuées au pire des cas par une recherche binaire dans un vecteur trié de taille N est exactement le nombre de bits dans la représentation binaire de N, c'est-à-dire $\lfloor \lg N \rfloor + 1$. Ces quantités sont solutions de la récurrence

$$B_N = B_{\lfloor N/2 \rfloor} + 1 \quad \forall N \geqslant 2$$

avec $B_1 = 1$.

Retour au tri-fusion

Récurrence donnant le nombre de copies effectuées par le tri-fusion :

$$C_N = C_{\lceil N/2 \rceil} + C_{\lfloor N/2 \rfloor} + N \quad \forall N \geqslant 2; \quad C_1 = 0 \,.$$

Soustraction de la récurrence N + 1 et N :

$$\begin{array}{rcl}
C_{N+1} & = & C_{\lfloor (N+1)/2 \rfloor} + C_{\lceil (N+1)/2 \rceil} + N + 1 \\
C_N & = & C_{\lfloor N/2 \rfloor} + C_{\lceil N/2 \rceil} + N \\
\hline
C_{N+1} - C_N & = & C_{\lfloor (N+1)/2 \rfloor} - C_{\lceil N/2 \rceil} + C_{\lceil (N+1)/2 \rceil} - C_{\lceil N/2 \rceil} + 1 \\
& = & C_{\lceil (N+1)/2 \rceil} - C_{\lfloor N/2 \rfloor} + 1 \\
& = & C_{\lfloor N/2 \rfloor + 1} + C_{\lfloor N/2 \rfloor} + 1
\end{array}$$

Résulte du fait que $\lceil (N+1)/2 \rceil = \lfloor N/2 \rfloor$ pour tout entier N

Conséquence de l'identité $\lceil (N+1)/2 \rceil = \lfloor N/2 \rfloor$+1, valable pour tout entier N.

En posant $D_N := C_{N+1} - C_N$, on obtient la récurrence $D_N = D_{\lfloor N/2 \rfloor} + 1$ pour $N \geqslant 2$ et $D_1 = C_2 - C_1 = 2 - 0 = 2$. Donc on obtient $D_N = \lfloor \lg N \rfloor + 2$ pour $N \geqslant 1$. Par conséquent,

$$\begin{aligned}
C_N &= C_N - 0 \\
&= C_N - C_1 \\
&= (C_N - C_{N-1}) + (C_{N-1} - C_{N-2}) + \cdots + (C_2 - C_1) \\
&= D_{N-1} + D_{N-2} + \cdots + D_1 \\
&= \sum_{k=1}^{N-1} (\lfloor \lg k \rfloor + 2) \\
&= (N-1) + \sum_{k=1}^{N-1} (\lfloor \lg k \rfloor + 1) \,.
\end{aligned}$$

Cette dernière quantité est N – 1 plus le nombre total de bits dans les représentations binaires des nombres entiers de 1 à N – 1.

Exemple

Par exemple, $C_9 = 8 + 21 = 29$ *car le nombre total de bits écrits quand on écrit les représentations binaires de* 1, 2, ..., 8 *est exactement* 21.

$$\left.\begin{array}{r} 1 \\ 10 \\ 11 \\ 100 \\ 101 \\ 110 \\ 111 \\ 1000 \end{array}\right\} \longrightarrow 21 \text{ bits}$$

Théorème

Théorème 14

Le nombre de copies effectuées par le tri fusion pour un vecteur de taille N *est exactement*

$$C_N = N\lfloor \lg N \rfloor + 2N - 2^{\lfloor \lg N \rfloor + 1} .$$

Ce nombre est une majoration sur le nombre de comparaisons effectuées par le tri fusion.

Rappels sur les comportements asymptotiques

Considérons des fonctions $f, g : \mathbb{N} \to \mathbb{R}_+$ s'annulant en un nombre fini de valeurs. Alors on écrit

- $f \sim g$ si $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$, on dit alors que f et g sont **asymptotiquement équivalentes** ;
- $f = O(g)$ s'il existe une constante $C > 0$ et $n_0 \in \mathbb{N}$ tels que $f(n) \leqslant Cg(n)$ pour $n \geqslant n_0$;
- $f = o(g)$ si $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$;
- $g = \Omega(f)$ si $f = O(g)$;
- $g = \omega(f)$ si $f = o(g)$;
- $f = \Theta(g)$ si $f = O(g)$ et $g = O(f)$, on dit alors que f et g ont même **comportement asymptotique**.

Exemple 15

Dans la liste des fonctions suivantes, chacune des fonctions est un $o(\cdot)$, *et donc un* $O(\cdot)$, *de la précédente :*

$$n^n,\ 2^n,\ n^2,\ n,\ \sqrt{n},\ \log^2 n,\ \log n,\ \log\log n .$$

De plus, $n^n = \omega(2^{\log^2 n})$ *et également* $2^n = \omega(2^{\log^2 n})$.

Générales

En analyse d'algo, on chercher à majorer le coût en temps d'un algo qui résout un problème de taille N en

- Produisant α sous-problèmes de tailles $\lfloor N/\beta \rfloor$ ou $\lceil N/\beta \rceil$
- Résolvant ces sous-problèmes de manière récursive
- Recombinant les solutions des sous-problèmes pour trouver une solution du problème original

Coût de construction des α sous-problèmes et de recombinaison des solutions correspondantes est majoré par une fonction f(N). On veut étudier la récurrence:

$$a_N = \alpha a_{N/\beta} + f(N) \quad \forall N \in \mathbb{N}\,, \qquad (3)$$

$a_N = 2a(N/2) + N$

où N/β doit être interprété tantôt comme $\lfloor N/\beta \rfloor$, tantôt comme $\lceil N/\beta \rceil$.

La récurrence pour le tri-fusion est un exemple où α = 2, β = 2 et f(N) = N. La recherche binaire est un autre exemple, cette fois avec α = 1, β = 2 et f(N) = 1.

$a_N = a(N/2) + 1$

Pour commencer, étudions l'équation fonctionnelle :

a(x) = αa(x/β) + x   ∀x > 1; a(x) = 0 ∀ <= 1   (4)

Ici on a α, β ∈ R tels que α > 0 et β > 1.

Théorème

Théorème 16

Si la fonction $a : \mathbb{R}_+ \to \mathbb{R}_+$ *est une solution de* (4), *alors :*

Cas 1. *Si* $\alpha > \beta$, *alors* $a(x) = \Theta(x^{\log_\beta \alpha})$

Cas 2. *Si* $\alpha = \beta$, *alors* $a(x) \sim x \log_\beta x = \Theta(x \log_2 x)$

Cas 3. *Si* $\alpha < \beta$, *alors* $a(x) \sim \frac{\beta}{\beta - \alpha} x = \Theta(x)$

Comportement asymptotique

Master Théorème

La généralisation de ce dernier résultat est le contenu du "Master Theorem"

Théorème 17 ("Master Theorem")

Soient $\alpha \geqslant 1$, $\beta > 1$ des constantes et $f : \mathbb{N} \to \mathbb{R}_+$ une fonction. Pour une suite $(a_N)_{N\in\mathbb{N}}$ solution de la récurrence diviser-pour-régner (3) :

Cas 1. *Si $f(N) = O(N^{\log_\beta \alpha - \varepsilon})$ pour $\varepsilon > 0$, alors $a_N = \Theta(N^{\log_\beta \alpha})$.*

Cas 2. *Si $f(N) = \Theta(N^{\log_\beta \alpha})$, alors $a_N = \Theta(N^{\log_\beta \alpha} \log_2 N)$.*

Cas 3. *Si $f(N) = \Omega(N^{\log_\beta \alpha + \varepsilon})$ pour $\varepsilon > 0$, et si $\alpha f(N/\beta) \leqslant Cf(N)$ pour une certaine constante $C < 1$ et N suffisamment grand, alors $a_N = \Theta(f(N))$.*

□

Quelques exemples sur le wikipedia: https://fr.wikipedia.org/wiki/Master_theorem

Simplification

$f = O(g) \to$ f est battu par g

$f = \Omega(g) \to$ f bat g

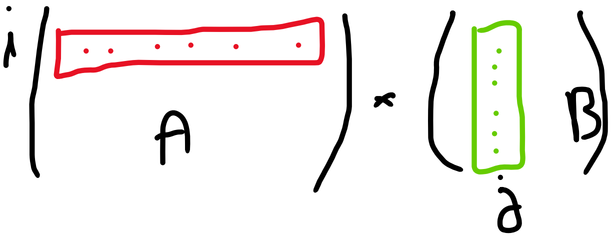
$f = \Theta(g) \to$ Les deux ensembles

Produit matriciel selon Strassen

Une vidéo sur le produit matriciel est dispo sur l'uv:
https://www.youtube.com/watch?v=aKhhYguY0DQ&ab_channel=KhanAcademy

Une en français :
https://www.youtube.com/watch?v=hQbIg6bmPhU&ab_channel=Lesmathsparl%27exemple

Pour calculer le produit de deux matrices, de taille n x n (le problème est alors de taille n).7



A = (a_{ij})

B = (b_{ij})

i représente les lignes et j les colonnes (dans le dessins ci-dessus on illustre la façon de faire un produit matriciel).

I,j ∈ [n]. Pour rappel, la matrice produit C = AB s'obtient comme suit : $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$.

"Le coefficient cij situé sur la i-ème ligne et j-ème colonne de C est le produit scalaire de la i-ème ligne de A et j-ème colonne de B."

Exemple de fonctionnement d'un algo classique

Pour $n = 2$, on a :

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

et l'algorithme classique calcule

$$\begin{aligned} c_{11} &= a_{11}b_{11} + a_{12}b_{21} \\ c_{12} &= a_{11}b_{12} + a_{12}b_{22} \\ c_{21} &= a_{21}b_{11} + a_{22}b_{21} \\ c_{22} &= a_{21}b_{12} + a_{22}b_{22} \ . \end{aligned}$$

Selon Strassen

Strassen a observé qu'en calculant les 7 produits

$$\begin{aligned} I &= (a_{11} - a_{22}) \cdot (b_{21} + b_{22}) \\ II &= (a_{11} + a_{22}) \cdot (b_{11} + b_{22}) \\ III &= (a_{11} - a_{21}) \cdot (b_{11} + b_{12}) \\ IV &= (a_{11} + a_{12}) \cdot b_{22} \\ V &= a_{11} \cdot (b_{12} - b_{22}) \\ VI &= a_{22} \cdot (b_{21} - b_{11}) \\ VII &= (a_{21} + a_{22}) \cdot b_{11} \end{aligned}$$

la matrice $C = AB$ peut s'obtenir comme suit :

$$\begin{aligned} c_{11} &= I + II - IV + VI \\ c_{12} &= IV + V \\ c_{21} &= VI + VII \\ c_{22} &= II - III + V - VII \ . \end{aligned}$$

Je skip les exemples. (Je remets tel quel car j'ai du mal de comprendre).

Observons que ce résultat n'utilise *pas* la commutativité du corps $(\mathbb{R}_{+,\cdot})$. Les formules restent vraies dans tout anneau, et en particulier dans l'anneau des matrices $k \times k$ réelles, $M_{k\times k}(\mathbb{R})$.

Pour $n > 2$, l'algorithme de Strassen découpe chaque matrice en quatre sous-matrices $\frac{n}{2} \times \frac{n}{2}$ (si n est impair, les tailles sont $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$, $\lfloor n/2 \rfloor \times \lceil n/2 \rceil$, $\lceil n/2 \rceil \times \lfloor n/2 \rfloor$ et $\lceil n/2 \rceil \times \lceil n/2 \rceil$) :

$$\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right) = \left(\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right) .$$

De nouveau, on a (exercice) :

$$\begin{aligned} C_{11} &= A_{11}B_{11} + A_{12}B_{21} \\ C_{12} &= A_{11}B_{12} + A_{12}B_{22} \\ C_{21} &= A_{21}B_{11} + A_{22}B_{21} \\ C_{22} &= A_{21}B_{12} + A_{22}B_{22} . \end{aligned}$$

Donc les formules de Strassen s'appliquent, et on peut calculer C_{11}, C_{12}, C_{21}, C_{22} en effectuant 7 produits matriciels et 18 additions matricielles. L'algorithme de Strassen effectue les additions matricielles directement et les produits matriciels récursivement.

Soit $T(n)$ le nombre total d'opérations arithmétiques (multiplications et additions de nombres réels) pour calculer le produit de deux matrices $n \times n$. On trouve la récurrence diviser-pour-régner suivante :

$$T(n) = 7 \cdot T(n/2) + 18 \cdot n^2.$$

Pour trouver le comportement asymptotique de $T(n)$, on utilise le "Master Theorem" avec $\alpha = 7$, $\beta = 2$ et $f(n) = 18n^2$.

On commence par le calcul $\log_\beta \alpha = 2,81\ldots > 2$. Par conséquent,

$$f(n) = O(n^2) = O(n^{\log_\beta \alpha - \varepsilon})$$

et on est donc dans le Cas 1 (prendre par exemple ε tel que $\log_\beta \alpha - \varepsilon = 2,81$). Par le "Master Theorem", on trouve

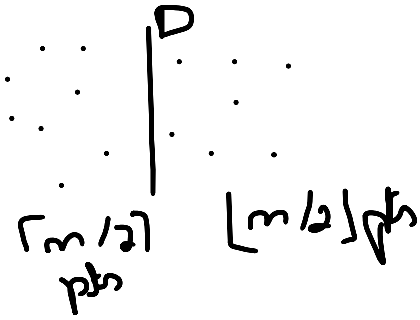
$$T(n) = \Theta(n^{\log_\beta \alpha}) = \Theta(n^{2,81\ldots}) .$$

L'algorithme de Strassen est donc (asymptotiquement) meilleur que l'algorithme classique !

Recherche d'une plus proche paire

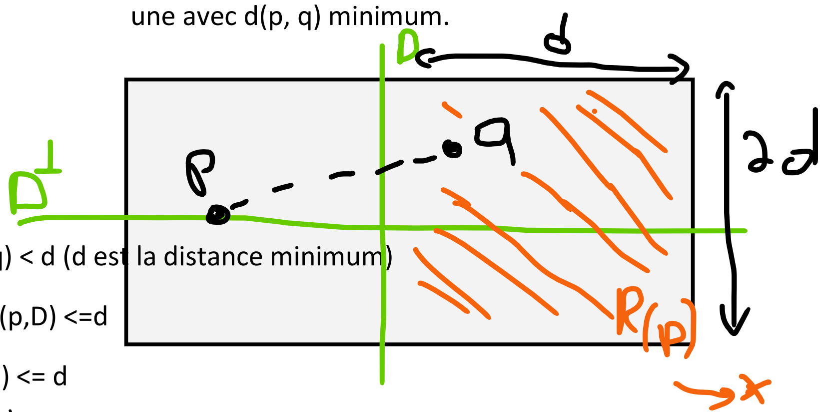
Etant donné n points dans le plan, comment trouver une paire de points séparés par une distance minimale ?

"L'algo trivial consistant à considérer chaque paire de points est en $\Theta(n^2)$." Pour faire mieux on commence par diviser le problème en deux sous problème de taille $\lfloor n/2 \rfloor$ et $\lceil n/2 \rceil$ (diviser-pour-régner), récurser et recombiner.



Comment aborder le problème :

1. Choisir une verticale D ne contenant aucun des points et divisant le nuage en deux
2. Comment recombiner ?
   a. Soit d en distance minimum observée à gauche ou à droite lors des appels récursifs. Pour résoudre le problème original, nous devons voir s'il existe une paire p, q de points avec p à gauche, q à droite et d(p, q) < d, et s'il existe un telle paire en trouver une avec d(p, q) minimum.



D(p,q) < d (d est la distance minimum)

⇔ d (p,D) <=d

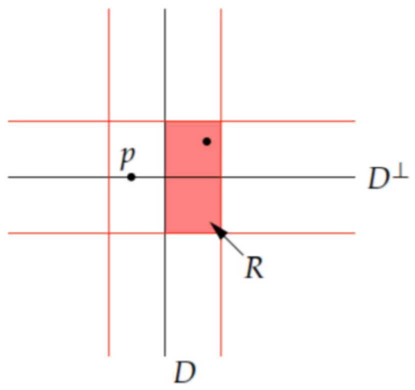
d(x,D) <= d

d(x,$D^\perp$)c <= d

Quelques observations :

1. On peut ignorer les points à distance $> d$ de la verticale D car tout point situé à une distance $> d$ de la droite D est situé à une distance $> d$ des points de l'autre côté de D.
2. On a $d(p, q) < d$ pour une paire p, q seulement si $d(p, D) \leqslant d$ et q se trouve dans un rectangle $2d \times d$ déterminé par p (voir dessin ci-dessous). Notons $D^\perp$ la perpendiculaire à D passant par p. Les seuls points q qui nous intéressent sont ceux tels que $d(q, D) \leqslant d$ et $d(q, D^\perp) \leqslant d$. L'ensemble de tous les points x tels que $d(x, D) \leqslant d$, $d(x, D^\perp) \leqslant d$ et x est situé à droite de D est un rectangle $2d \times d$ que nous noterons $R = R(p)$.

Même dessin mais plus propre



Combien de points q peut-on trouver dans le rectangle R ? Nous savons bien sûr que deux points q, q' dans le rectangle R figurant parmi les n points donnés en entrée vérifient d(q, q') > d. Il résulte de ceci que le nombre de points q figurant parmi les n points donnés et appartenant au rectangle R est borné. Intuitivement, on dirait que le maximum est 6. Vérifions qu'il y a au plus 10 tels points q.

Lemma

Lemma 5

Soient $d > 0$, R un rectangle $\overbrace{2d \times d}^{R(p)}$ et $X \subseteq R$ un ensemble de points tels que $d(q, q') \geqslant d$ pour tous $q, q' \in X$ distincts. Alors $|X| \leqslant 10$.

T(n) = la complexité en temps de la partie récursive de l'algo pour une instance de taille n.

- Précalcule : Construire deux listes triées :
  - Une qui contient l'ensemble des points triés par abscisses croissantes
  - L'autre, ensemble des points tirés par ordonnées croissantes. [Θ(n lg n)]
- Partie récursive :
  - Trouver une droite verticale D divisant l'ensemble des points en deux parties de tailles $\lfloor n/2 \rfloor$ et $\lceil n/2 \rceil$. [Θ(n)]
  - Déterminer récursivement une paire la plus proche dans chacune des parties (gauche et droite), soit d la distance min observée. [T(dn/2e) + T(bn/2c)]
  - Pour tous les points p à gauche de D et à distance <= d de D, inspecter tous les points q à droite de D, dans le rectangle R(p) défini par p, retenir la distance minimum si celle-ci est < d, et une paire à distance minimum. [Θ(n)]
  - Retourner la distance minimum et une paire de points à distance minimum. [Θ(1)]

On trouve alors la récurrence diviser-pour-régner suivante: T(n) = 2T(n/2) + Θ(n), qui a pour solution T(n) = Θ(n $\log_2$ n) .

A cela on rajoute Θ(n $\log_2$ n) pour le précalcule. Au total, le problème peut être résolu en O(n $\log_2$ n).

Autres exemples de récurrence

Celles-ci je les remets vite fait.

Calcul de racine carrée

Théorème

Théorème 19

Pour tout $\beta > 0$, la solution $(a_n)_{n\in\mathbb{N}}$ de la récurrence (6) converge vers $\sqrt{\beta}$.

La récurrence (6) en question :

$$a_n = \frac{1}{2}\left(a_{n-1} + \frac{\beta}{a_{n-1}}\right) \quad n \geq 1; \quad a_0 = 1 \ .$$

Fractions continuées

Etude de la récurrence non linéaire d'ordre 1 suivante :

$$a_n = \frac{1}{1 + a_{n-1}} \quad \forall n \geqslant 1; \quad a_0 = 1 \ .$$

Aha ! On dirait que les valeurs obtenues sont des quotients de nombres de Fibonacci consécutifs. (voir début du chapitre).

$$\lim_{n\to\infty} a_n = \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}}} \ .$$

On appelle ce type de fraction une **fraction continuée**. De telles expressions se révèlent utiles pour approximer des nombres (ir)rationnels par des rationnels dont la taille, c'est-à-dire le nombre de bits total dans la représentation du rationnel en base 2, est petite.

Liste de théorèmes du chapitre

À completer

Bonus

Vidéo pour apprendre à faire des démos de récurrences (je l'ai pas encore regardé mais je vous la mets quand même jsp ce que ça vaut):
https://www.youtube.com/watch?v=udGGlHdSAgc&ab_channel=YvanMonka