

Clôture transitive des graphes et tous les plus courts chemins

301

Clôture transitive (accessibilité)

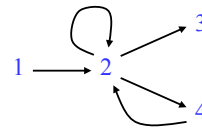
Problème

$G = (S, A)$ graphe (orienté)

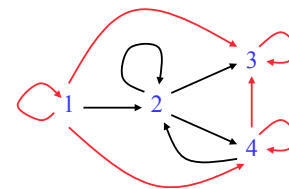
Calculer $H = (S, B)$ où B est la clôture réflexive et transitive de A .

Note : $(s,t) \in B$ ssi il existe un chemin de s à t dans G

graphe G



graphe H



302

Représentations matricielles

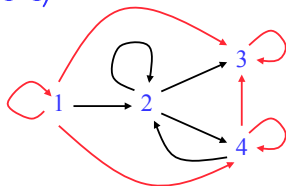
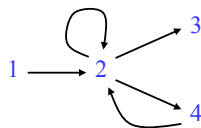
Matrices $n \times n$ où $n = \text{card } S$

A matrice d'adjacence de G
= matrice des chemins de longueur 1

B matrice d'adjacence de H
= matrice des chemins de G

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$



303

Clôture par produits

Notation

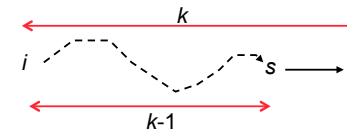
A_k = matrice des chemins de longueur k dans G

A_0 = I (matrice identité)

A_1 = matrice des chemins de longueur 1 = A

Lemme

Pour tout $k \geq 0$, $A_k = A^k$
(produit booléen)



Preuve

$A_k[i,j] = 1$ ssi $\exists s \in S$ $A_{k-1}[i,s] = 1$ et $A[s,j] = 1$

soit $A_k[i,j] = \sum_s A_{k-1}[i,s] \cdot A[s,j]$ (Σ somme booléenne)

soit $A_k = A_{k-1} \cdot A$ et $A_0 = I$

donc $A_k = A^k$

304

Clôture par produits (suite)

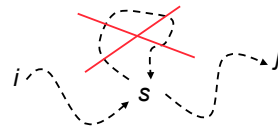
UMLV ©

Chemin simple :

chemin qui passe une seule fois par chacun de ses sommets

Lemme

- ∃ chemin de i à j dans G ssi
- ∃ chemin simple de i à j dans G



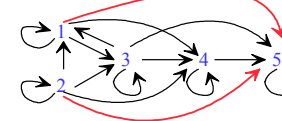
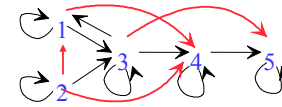
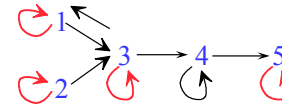
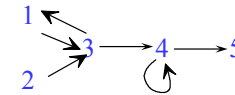
$B[i,j] = 1$ ssi ∃ chemin de i à j dans G
 ssi ∃ chemin simple de i à j dans G
 ssi ∃ k $0 \leq k \leq \text{card}S - 1$ $A_k[i,j] = 1$
 ssi ∃ k $0 \leq k \leq \text{card}S - 1$ $A^k[i,j] = 1$
 donc $B = I + A + A^2 + \dots + A^{\text{card}S - 1}$

Calcul de B par schéma de Horner:

$$A_0 = I$$

$$A_i = I + A_{i-1} \cdot A \text{ pour } i=1.. \text{card}S - 1$$

305



3 produits de matrices

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$I + A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$I + A + A^2 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= I + (I + A) \cdot A =$$

$$I + A + A^2 + A^3 = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= I + (I + A + A^2) \cdot A =$$

$$= I + A + A^2 + A^3 + A^4$$

$$= I + (I + A + A^2 + A^3) \cdot A = B$$

UMLV ©

306

Complexité de l'algorithme

UMLV ©

$n = \text{card } S$

$n-1$ additions et $n-1$ multiplications de matrices booléennes $n \times n$
 $\Rightarrow O(n \cdot M(n))$

chaque multiplication se fait en $O(n^3)$ opérations $\Rightarrow O(n^4)$

il existe des algorithmes de calcul de la multiplication des matrices booléennes en temps $o(n^3)$:
 Strassen (1969): $O(n^{2,81})$

307

Complexité de l'algorithme

UMLV ©

$n = \text{card } S$

$n-1$ additions et $n-1$ multiplications de matrices booléennes $n \times n$
 $\Rightarrow O(n \cdot M(n))$

chaque multiplication se fait en $O(n^3)$ opérations $\Rightarrow O(n^4)$

il existe des algorithmes de calcul de la multiplication des matrices booléennes en temps $o(n^3)$:
 Strassen (1969): $O(n^{2,81})$

L'algorithme n'est pas optimal !

Définir $v(i,j)=1$ pour chaque arc $(i,j) \in A$

Pour chaque sommet i exécuter l'algorithme de Dijkstra avec i comme source

$B[i,j]=1$ ssi $\delta(i,j) < \infty$

Temps d'exécution $O(n \cdot n^2) = O(n^3)$

308

Autre récurrence

UMLV ©

Notation

B_k = matrice des chemins de longueur $\leq k$ dans G

$B_0 = I$ (matrice identité)

$B_1 =$ matrice des chemins de longueur $\leq 1 = I + A$

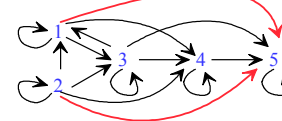
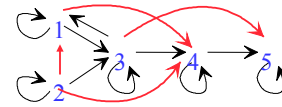
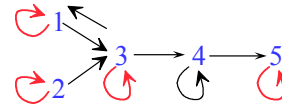
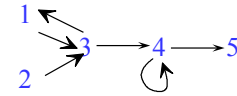
$B_{n-1} =$ matrice des chemins simples $= B$

Lemme: $B_k = B_{k-1} \cdot (I + A)$

\Rightarrow Pour tout $k \geq 1$, $B_k = (I + A)^k$ et en particulier $B_{2k} = B_k \cdot B_k$

Calcul de B comme une puissance $n-1$ en temps $O(\log n \cdot M(n)) = O(\log n \cdot n^3)$

309



2 produits de matrices

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B_1 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B_2 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B = B_4 = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

UMLV ©

310

Algorithme de Warshall (Roy-Warshall)

UMLV ©

$G = (S, A)$ avec $S = \{1, 2, \dots, n\}$

Chemin dans $G : i \rightarrow s_1 \rightarrow s_2 \dots s_{m-1} \rightarrow j$

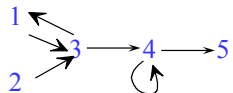
Sommets intermédiaires : s_1, s_2, \dots, s_{m-1}

Notation

$C_k =$ matrice des chemins de G dont les sommets intermédiaires sont tous $\leq k$

$C_0 = I + A$

$C_n =$ matrice des chemins de $G = B$



Chemin de 2 à 4 : (2,3), (3,1), (1,3), (3,4), (4,4)

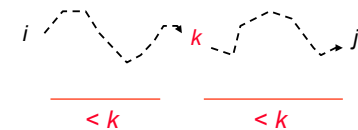
sommets intermédiaires : 1, 3, 4

311

Récurrence

UMLV ©

Chemin simple



Lemme Pour tout $k \geq 1$,

$$C_k[i,j] = 1 \text{ ssi } C_{k-1}[i,j] = 1 \text{ ou } (C_{k-1}[i,k] = 1 \text{ et } C_{k-1}[k,j] = 1)$$

Calcul

de C_k à partir de C_{k-1} en temps $O(n^2)$

de $B = C_n$ en temps $O(n^3)$

312

UMLV ©

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C_0 = C_1 = C_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B = C_4 = C_5 = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

~1 produit de matrice

313

UMLV ©

fonction clôture (graphe $G = (S, A)$) : matrice ;

début

$n \leftarrow \text{card } S$;

pour $i \leftarrow 1$ à n **faire**

pour $j \leftarrow 1$ à n **faire**

si $i=j$ ou $A[i,j] = 1$ **alors**

$B[i,j] \leftarrow 1$;

sinon

$B[i,j] \leftarrow 0$;

pour $k \leftarrow 1$ à n **faire**

pour $i \leftarrow 1$ à n **faire**

pour $j \leftarrow 1$ à n **faire**

$B[i,j] \leftarrow B[i,j] + B[i,k] \cdot B[k,j]$;

retour B ;

fin

+ est la somme booléenne ; temps d'exécution $O(n^3)$

314

Bilan provisoire

UMLV ©

Trois algorithmes pour calculer la clôture transitive:

- polynôme matriciel : $O(n \cdot M(n)) = O(n^4)$
- puissance d'une matrice : $O(\log n \cdot M(n)) = O(\log n \cdot n^3)$
- algorithme de Warshall : $O(n^3)$

On va généraliser ces idées pour calculer tous les plus courts chemins dans un graphe valué

315

Distances

UMLV ©

$G = (S, A, v)$ graphe valué $S = \{1, 2, \dots, n\}$ $v : A \rightarrow \mathbf{N}$

Matrice des poids : $W = (w[i,j])$ avec

$$w[i,j] = \begin{cases} 0 & \text{si } i=j \\ v((i,j)) & \text{si } (i,j) \in A \\ \infty & \text{sinon} \end{cases}$$

Poids d'une suite $c = ((s_0, s_1), (s_1, s_2), \dots, (s_{k-1}, s_k))$ où les $s_i \in S$

$$w(c) = \sum w[s_{i-1}, s_i]$$

Distance de s à t

$$d(s, t) = \min\{w(c) \mid c \text{ suite de } s \text{ à } t\}$$

Plus court chemin de s à t :

chemin c , s'il existe, tel que $w(c) = d(s, t)$

316

Première méthode : produit de matrices

UMLV ©

Soit $d^{(m)}(i,j)$ la valeur minimale d'un chemin de i à j à condition que ce chemin contienne au plus m arcs.

$$d(i,j) = d^{(n)}(i,j)$$

Idee : Procéder par induction sur m

317

$$d^{(0)}(i,j) = \begin{cases} 0 & \text{si } i=j \\ \infty & \text{sinon} \end{cases}$$

Pour $m \geq 1$

$$d^{(m)}(i,j) = \min (d^{(m-1)}(i,j), \min\{d^{(m-1)}(i,k) + w_{kj} \mid 1 \leq k \leq n\}) = \min\{d^{(m-1)}(i,k) + w_{kj} \mid 1 \leq k \leq n\}$$

En terme de matrices on obtient $D^{(m)} = D^{(m-1)} \cdot W$ où
 \min joue le rôle de $+$ et
 $+$ joue le rôle de \cdot .

En calculant $D = W^n$ par carrés successives, on obtient un algorithme avec complexité $O(n^3 \cdot \log n)$

UMLV ©

318

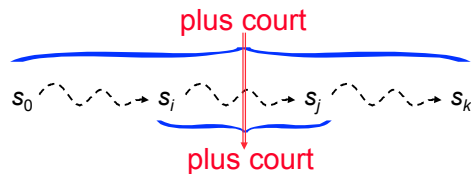
Méthode basée sur les sommets intermédiaires

UMLV ©

Lemme de base (rappel)

$((s_0, s_1), \dots, (s_i, s_{i+1}), \dots, (s_{j-1}, s_j), \dots, (s_{k-1}, s_k))$
 plus court chemin de s_0 à s_k dans G

$\Rightarrow ((s_i, s_{i+1}), \dots, (s_{j-1}, s_j))$ plus court chemin de s_i à s_j dans G



319

Algorithme de Floyd(-Warshall)

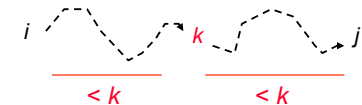
UMLV ©

Notation

$D_k = (D_k[i, j] \mid 1 \leq i, j \leq n)$ avec
 $D_k[i, j] = \min\{w(c) \mid c \text{ chemin de } i \text{ à } j \text{ dont les sommets intermédiaires sont tous } \leq k\}$

$$D_0 = W$$

$$D_n = \text{matrice des distances de } G = D$$



Lemme Pour tout $k \geq 1$,

$$D_k[i, j] = \min\{D_{k-1}[i, j], D_{k-1}[i, k] + D_{k-1}[k, j]\}$$

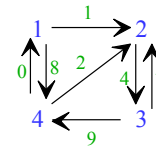
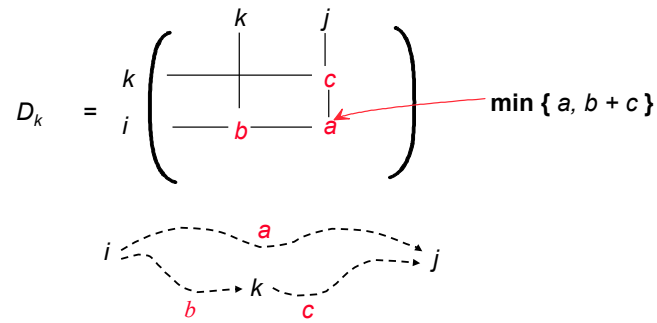
Calcul

de D_k à partir de D_{k-1} en temps $O(n^2)$

de $D = D_n$ en temps $O(n^3)$

320

pour $k \leftarrow 1$ à n faire
 pour $i \leftarrow 1$ à n faire
 pour $j \leftarrow 1$ à n faire
 $D[i, j] \leftarrow \min \{ D[i, j], D[i, k] + D[k, j] \};$



$$D_0 = W = \begin{pmatrix} 0 & 1 & \infty & 8 \\ \infty & 0 & 4 & \infty \\ \infty & 7 & 0 & 9 \\ 0 & 2 & \infty & 0 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 0 & 1 & \infty & 8 \\ \infty & 0 & 4 & \infty \\ \infty & 7 & 0 & 9 \\ 0 & 1 & \infty & 0 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0 & 1 & 5 & 8 \\ \infty & 0 & 4 & \infty \\ \infty & 7 & 0 & 9 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

$$D_3 = \begin{pmatrix} 0 & 1 & 5 & 8 \\ \infty & 0 & 4 & 13 \\ \infty & 7 & 0 & 9 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

$$D_4 = \begin{pmatrix} 0 & 1 & 5 & 8 \\ 13 & 0 & 4 & 13 \\ 9 & 7 & 0 & 9 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

Mémorisation des chemins

Mémorisation explicite des plus courts chemins de i à j , $1 \leq i, j \leq n$
 n^2 chemins de longueur maximale $n-1$: espace $O(n^3)$

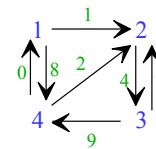
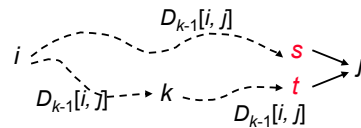
Matrice des prédécesseurs : espace $\Theta(n^2)$

$P_k = (P_k[i, j] \mid 1 \leq i, j \leq n)$ avec
 $P_k[i, j]$ = prédécesseur de j sur un plus court chemin de i à j dont
 les sommets intermédiaires sont tous $\leq k$

Récurrance

$$P_0[i, j] = \begin{cases} i & \text{si } (i, j) \in A \\ - & \text{sinon} \end{cases}$$

$$P_k[i, j] = \begin{cases} P_{k-1}[i, j] & \text{si } D_{k-1}[i, j] \leq D_{k-1}[i, k] + D_{k-1}[k, j] \\ P_{k-1}[k, j] & \text{sinon} \end{cases}$$



$$D_0 = W = \begin{pmatrix} 0 & 1 & \infty & 8 \\ \infty & 0 & 4 & \infty \\ \infty & 7 & 0 & 9 \\ 0 & 2 & \infty & 0 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 0 & 1 & \infty & 8 \\ \infty & 0 & 4 & \infty \\ \infty & 7 & 0 & 9 \\ 0 & 1 & \infty & 0 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} 0 & 1 & 5 & 8 \\ \infty & 0 & 4 & \infty \\ \infty & 7 & 0 & 9 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

$$D_3 = \begin{pmatrix} 0 & 1 & 5 & 8 \\ \infty & 0 & 4 & 13 \\ \infty & 7 & 0 & 9 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

$$D_4 = \begin{pmatrix} 0 & 1 & 5 & 8 \\ 13 & 0 & 4 & 13 \\ 9 & 7 & 0 & 9 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

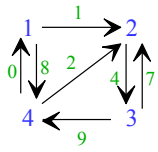
$$P_0 = \begin{pmatrix} - & 1 & - & 1 \\ - & - & 2 & - \\ - & 3 & - & 3 \\ 4 & 4 & - & - \end{pmatrix}$$

$$P_1 = \begin{pmatrix} - & 1 & - & 1 \\ - & - & 2 & - \\ - & 3 & - & 3 \\ 4 & 1 & - & - \end{pmatrix}$$

$$P_2 = \begin{pmatrix} - & 1 & 2 & 1 \\ - & - & 2 & - \\ - & 3 & - & 3 \\ 4 & 1 & 2 & - \end{pmatrix}$$

$$P_3 = \begin{pmatrix} - & 1 & 2 & 1 \\ - & - & 2 & 3 \\ - & 3 & - & 3 \\ 4 & 1 & 2 & - \end{pmatrix}$$

$$P_4 = \begin{pmatrix} - & 1 & 2 & 1 \\ 4 & - & 2 & 3 \\ 4 & 3 & - & 3 \\ 4 & 1 & 2 & - \end{pmatrix}$$



$$D_4 = \begin{pmatrix} 0 & 1 & 5 & 8 \\ 13 & 0 & 4 & 13 \\ 9 & 7 & 0 & 9 \\ 0 & 1 & 5 & 0 \end{pmatrix}$$

$$P_4 = \begin{pmatrix} - & 1 & 2 & 1 \\ 4 & - & 2 & 3 \\ 4 & 3 & - & 3 \\ 4 & 1 & 2 & - \end{pmatrix}$$

Exemple de chemin

distance de 2 à 1 = $D_4[2,1] = 13$

$P_4[2,1] = 4$; $P_4[2,4] = 3$; $P_4[2,3] = 2$;

