# Chemins dans un graphe

## Approche algébrique

Jean-Marc.Vincent@univ-grenoble-alpes.fr[1]

[1] Laboratoire LIG
Équipe-Projet INRIA POLARIS
Université Grenoble-Alpes

Algorithmique et Modélisation
L3-INFO

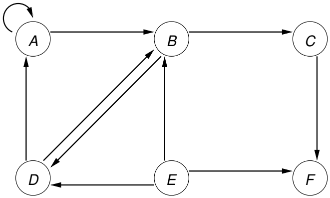
# Le contexte

Soit $\mathcal{G} = (\mathcal{X}, \mathcal{A})$ un graphe orienté.
$\mathcal{X} = \{x_1, \cdots, x_n\}$ ensemble de $n$ sommets, $|\mathcal{X}| = n$
$\mathcal{A}$ l'ensemble des arcs du graphe, on notera $|\mathcal{A}| = m$

## Problèmes

- Pour $x$ et $y$ sommets du graphe, existe-t'il un chemin de longueur $k$ de $x$ à $y$ ?
- Pour tous couple $x$ et $y$ sommets du graphe, existe-t'il un chemin de $x$ à $y$ ?

**Exemple**




**Questions**

- Existe-t'il un chemin de longueur
  2 de $B$ à $D$ ?
  4 de $E$ à $A$ ?
  7 de $D$ à $F$ ?
- Pour tout couple de sommets $(x, y)$ déterminer s'il existe un chemin de $x$ à $y$ ?

- Existe-t'il un chemin de longueur 2 de $B$ à $D$ ?

- Existe-t'il un chemin de longueur
  2 de $B$ à $D$ ?

  non

  4 de $E$ à $A$ ?

- Existe-t'il un chemin de longueur
  2 de $B$ à $D$ ?

  > non

  4 de $E$ à $A$ ?

  > oui, par exemple le chemin [E,B,D,A,A] ou [E,D,B,D,A]

  7 de $D$ à $F$ ?

- Existe-t'il un chemin de longueur
  2 de $B$ à $D$ ?

  non

  4 de $E$ à $A$ ?

  oui, par exemple le chemin [E,B,D,A,A] ou [E,D,B,D,A]

  7 de $D$ à $F$ ?

  oui, par exemple le chemin [D,A,B,D,A,B,C,F]

- Pour tout couple de sommets $(x, y)$ déterminer s'il existe un chemin de $x$ à $y$ ?

- Existe-t'il un chemin de longueur
  2 de $B$ à $D$ ?

  non

  4 de $E$ à $A$ ?

  oui, par exemple le chemin [E,B,D,A,A] ou [E,D,B,D,A]

  7 de $D$ à $F$ ?

  oui, par exemple le chemin [D,A,B,D,A,B,C,F]

- Pour tout couple de sommets $(x, y)$ déterminer s'il existe un chemin de $x$ à $y$ ?

  Il existe un chemin de

  A à A,B,C,D,F
  B à A,B,C,D,F
  C à C,F
  D à A,B,C,D,F
  E à A,B,C,D,E,F
  F à aucun

  Remarque : on considère également les chemins de longueur 0
  c'est également le même problème que la recherche de composantes connexes

# REPRÉSENTATION DU GRAPHE

On représente le graphe par sa matrice d'adjacence $A = ((a_{x,y}))_{(x,y)\in\mathcal{X}^2}$, c'est à dire un tableau à 2 dimensions de valeurs booléennes avec la sémantique

$$a_{x,y} = \begin{cases} \text{Vrai} & \text{s'il y a un arc de } x \text{ à } y \\ \text{Faux} & \text{sinon} \end{cases}$$

par abus de notation indiquera Vrai et Faux par des variables booléennes respectivement 1 et 0

**Question**

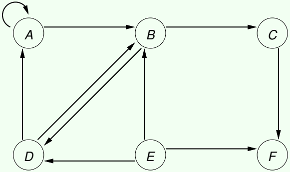
- Écrire la matrice d'adjacence associée à l'exemple.

- Écrire la matrice d'adjacence associée à l'exemple.

- Écrire la matrice d'adjacence associée à l'exemple.

$$A = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 1 & 1 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 1 & 1 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 1 \\ D & 1 & 1 & 0 & 0 & 0 & 0 \\ E & 0 & 1 & 0 & 1 & 0 & 1 \\ F & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Graphe (pour mémoire)



# CHEMINS DE LONGUEUR 2

On définit le graphe associé aux chemins de longueur 2 comme le graphe sur $\mathcal{X}$ dont la matrice d'adjacence $A^{(2)}$ est définie par

$$a^{(2)}_{x,y} = \begin{cases} \text{Vrai} & \text{s'il existe un chemin de longueur 2 de } x \text{ à } y \\ \text{Faux} & \text{sinon} \end{cases}$$

**Questions**

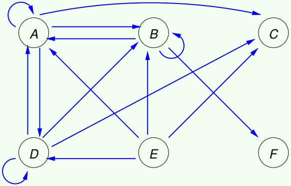
- Dessiner le graphe associé aux chemins de longueur 2 de l'exemple
- Écrire la matrice d'adjacence associée aux chemins d'e longueur 2 de l'exemple.
- Écrire un algorithme qui prend en argument la matrice d'adjacence d'un graphe et retourne la matrice d'adjacence associée au chemins de longueur 2

- Écrire la matrice d'adjacence associée à l'exemple.

- Écrire la matrice d'adjacence associée à l'exemple.

$$A^{(2)} = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 1 & 1 & 1 & 1 & 0 & 0 \\ B & 1 & 1 & 0 & 0 & 0 & 1 \\ C & 0 & 0 & 0 & 0 & 0 & 0 \\ D & 1 & 1 & 1 & 1 & 0 & 0 \\ E & 1 & 1 & 1 & 1 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Graphe des chemins de longueur 2




chemin2($A$)

**Données:** matrice d'adjacence d'un graphe de taille $n$

**Résultat:** matrice d'adjacence du graphe des chemins de longueur 2

A2 = matrice booléenne $n \times n$ initialisée à faux (0)

**for** $i = 1$ **to** $n$

  **for** $j = 1$ **to** $n$

    **for** $k = 1$ **to** $n$

      $a2_{i,j} = a2_{i,j} \vee a_{i,k} \wedge a_{k,j}$

`Return` A2

# CHEMINS DE LONGUEUR 3

On définit le graphe associé aux chemins de longueur 3 comme le graphe sur $\mathcal{X}$ dont la matrice d'adjacence $A^{(3)}$ se définit de la même manière que l'on a défini $A^{(2)}$.

## Questions

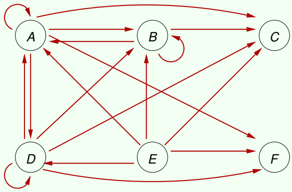
- Dessiner le graphe associé aux chemins de longueur 3 de l'exemple
- Écrire la matrice d'adjacence associée aux chemins de longueur 3 de l'exemple.
- Quelle opération sur des matrices d'adjacence pouvez-vous construire pour calculer plus synthétiquement $A^{(3)}$ ?
- Écrire un algorithme qui réalise cette opération.
- En déduire un algorithme qui calcule $A^{(k)}$ matrice d'adjacence des chemins de longueur $k$

- Dessiner le graphe associé aux chemins de longueur 3 de l'exemple
- Écrire la matrice d'adjacence associée aux chemins de longueur 3 de l'exemple.

- Dessiner le graphe associé aux chemins de longueur 3 de l'exemple
- Écrire la matrice d'adjacence associée aux chemins de longueur 3 de l'exemple.

$$A^{(3)} = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 1 & 1 & 1 & 1 & 0 & 1 \\ B & 1 & 1 & 1 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 \\ D & 1 & 1 & 1 & 1 & 0 & 1 \\ E & 0 & 1 & 0 & 1 & 0 & 1 \\ F & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Graphe des chemins de longueur 3



- Quelle opération sur des matrices d'adjacence pouvez-vous construire pour calculer plus synthétiquement $A^{(3)}$ ?
- Écrire un algorithme qui réalise cette opération.

- Quelle opération sur des matrices d'adjacence pouvez-vous construire pour calculer plus synthétiquement $A^{(3)}$ ?
- Écrire un algorithme qui réalise cette opération.

Il s'agit du produit de matrice avec des booléens

```
Produit (A, B)
    Données: A, B matrice booléennes n × n
    Résultat: matrice booléenne égale à A ⊗ B

    C = matrice booléenne n × n initialisée à faux (0)
    for i = 1 to n
        for j = 1 to n
            for k = 1 to n
                c_{i,j} = c_{i,j} ∨ a_{i,k} ∧ b_{k,j}
    Return C
```

- En déduire un algorithme qui calcule $A^{(k)}$ matrice d'adjacence des chemins de longueur $k$

- Quelle opération sur des matrices d'adjacence pouvez-vous construire pour calculer plus synthétiquement $A^{(3)}$ ?
- Écrire un algorithme qui réalise cette opération.

Il s'agit du produit de matrice avec des booléens

```
Produit (A, B)
    Données: A, B matrice booléennes n × n
    Résultat: matrice booléenne égale à A ⊗ B

    C = matrice booléenne n × n initialisée à faux (0)
    for i = 1 to n
        for j = 1 to n
            for k = 1 to n
                c_{i,j} = c_{i,j} ∨ a_{i,k} ∧ b_{k,j}
    Return C
```

- En déduire un algorithme qui calcule $A^{(k)}$ matrice d'adjacence des chemins de longueur $k$

Il s'agit simplement d'élever la matrice $A$ à la puissance $k$ (voir le cours sur l'exponentiation)

# Représentation algébrique

Vous l'avez bien compris, il s'agit du produit de matrices, sauf que l'opérateur $+$ a été remplacé par l'opérateur "ou" noté $\vee$ et l'opérateur "x" par l'opérateur "et" noté $\wedge$.
On peut donc écrire

$$a_{x,y}^{(k+1)} = \bigvee_{z} a_{x,z}^{(k)} \wedge a_{z,y}. \tag{1}$$

## Questions

- Expliquer cette équation en une phrase
- Expliquer pourquoi l'on a également

$$a_{x,y}^{(k+1)} = \bigvee_{z} a_{x,z} \wedge a_{z,y}^{(k)}. \tag{2}$$

- Faire un dessin explicatif de cette équation

- Expliquer cette équation en une phrase

- Expliquer cette équation en une phrase

Il existe un chemin de longueur $k+1$ si et seulement si il existe un sommet $z$ tel que il existe un chemin de longueur $k$ de $x$ à $z$ et un arc de $z$ à $y$ (le dernier arc du chemin de longueur $k+1$ de $x$ à $y$

- Expliquer pourquoi l'on a également

$$a_{x,y}^{(k+1)} = \bigvee_z a_{x,z} \wedge a_{z,y}^{(k)}.$$

- Expliquer cette équation en une phrase

Il existe un chemin de longueur $k+1$ si et seulement si il existe un sommet $z$ tel que il existe un chemin de longueur $k$ de $x$ à $z$ et un arc de $z$ à $y$ (le dernier arc du chemin de longueur $k+1$ de $x$ à $y$

- Expliquer pourquoi l'on a également

$$a^{(k+1)}_{x,y} = \bigvee_z a_{x,z} \wedge a^{(k)}_{z,y}.$$

C'est exactement le même raisonnement qu'avant, en regardant le premier arc du chemin de $x$ à $y$

- Faire un dessin explicatif de cette équation

- Expliquer cette équation en une phrase

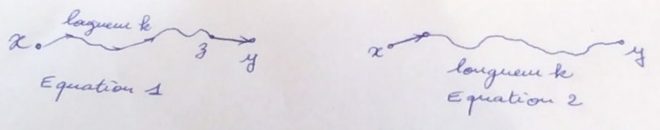
Il existe un chemin de longueur $k+1$ si et seulement si il existe un sommet $z$ tel que il existe un chemin de longueur $k$ de $x$ à $z$ et un arc de $z$ à $y$ (le dernier arc du chemin de longueur $k+1$ de $x$ à $y$

- Expliquer pourquoi l'on a également

$$a_{x,y}^{(k+1)} = \bigvee_z a_{x,z} \wedge a_{z,y}^{(k)}.$$

C'est exactement le même raisonnement qu'avant, en regardant le premier arc du chemin de $x$ à $y$

- Faire un dessin explicatif de cette équation



# Représentation algébrique (2)

On est donc maintenant capable de résoudre le problème de l'existence d'un chemin entre tout couple de sommets
On définit la matrice d'adjacence du graphe d'existence d'un chemin par

$$A^* = A^{(0)} \vee A^{(1)} \vee A^{(2)} \vee A^{(3)} \vee \cdots \vee A^{(k)} \vee \cdots \tag{3}$$

où
- $A^{(1)}$ représente les chemins de longueur 1 donc c'est la matrice $A$ du graphe
- $A^{(0)}$ représente les chemins de longueur 0 donc c'est la matrice ayant que des 1 sur la diagonale et des 0 ailleurs
- $A^{(k)}$ représente les chemins de longueur $k$

**Questions**

- Expliquer cette équation 3 en une phrase
- Expliquer pourquoi il n'est pas nécessaire de faire ces $\vee$ jusqu'à l'infini, on peut s'arrêter à $n-1$
- Écrire l'algorithme de calcul de la matrice de l'existence des chemins.

Notation algébrique, de manière plus compacte on note

$$A^* = \bigvee_{k=0}^{n-1} A^{(k)}$$

- Expliquer cette équation 3 en une phrase

- Expliquer cette équation 3 en une phrase

> Il existe un chemin de $x$ à $y$ ssi il existe un chemin de longueur 1 ou un chemin de longueur 2 ou etc On fait le ou entre toutes les matrices

- Expliquer pourquoi il n'est pas nécessaire de faire ces $\vee$ jusqu'à l'infini, on peut s'arrêter à $n-1$

- Expliquer cette équation 3 en une phrase

Il existe un chemin de $x$ à $y$ ssi il existe un chemin de longueur 1 ou un chemin de longueur 2 ou etc On fait le ou entre toutes les matrices

- Expliquer pourquoi il n'est pas nécessaire de faire ces $\vee$ jusqu'à l'infini, on peut s'arrêter à $n-1$

S'il existe un chemin de longueur $k \geqslant n$ de $x$ à $y$ alors, comme le graphe n'a que $n$ sommets le chemin passe plusieurs fois par un même sommet (principe des tiroirs et des chaussettes). On peut donc trouver un chemin plus court de $x$ à $y$ ? par induction il en existe un de longueur inférieure strictement à $n$

- Écrire l'algorithme de calcul de la matrice de l'existence des chemins.

- Expliquer cette équation 3 en une phrase

Il existe un chemin de $x$ à $y$ ssi il existe un chemin de longueur 1 ou un chemin de longueur 2 ou etc On fait le ou entre toutes les matrices

- Expliquer pourquoi il n'est pas nécessaire de faire ces $\vee$ jusqu'à l'infini, on peut s'arrêter à $n-1$

S'il existe un chemin de longueur $k \geqslant n$ de $x$ à $y$ alors, comme le graphe n'a que $n$ sommets le chemin passe plusieurs fois par un même sommet (principe des tiroirs et des chaussettes). On peut donc trouver un chemin plus court de $x$ à $y$ ? par induction il en existe un de longueur inférieure strictement à $n$

- Écrire l'algorithme de calcul de la matrice de l'existence des chemins.

On écrit d'abord une fonction qui réalise la somme booléenne de 2 matrice puis on itère en faisant un cumul

```
Somme (A, B)
    Données: A, B matrice booléennes n × n
    Résultat: matrice booléenne égale à A ⊕ B

    C = matrice booléenne n × n
    for i = 1 to n
        for j = 1 to n
            c_{i,j} = a_{i,j} ∨ b_{i,j}
    Return C
```

Il s'agit d'un algorithme de fermeture reflexive-transitive d'un graphe

```
Fermeture (A)
    Données: A matrice d'adjacence d'un graphe n × n
    Résultat: matrice d'adjacence de la fermeture de ce graphe A*

    P = matrice booléenne n × n initialisée à 1 sur la diagonale 0 ailleurs
    F = matrice booléenne n × n initialisée à 1 sur la diagonale 0 ailleurs
    for i = 1 to n − 1
        P = Produit (P,A)                          // calcule A^(i)
        F = Somme (F,P)           // calcule A^(0) ∨ A^(1) ∨ · · · ∨ A^(i)
    Return F
```

# Analyse du coût

**Questions**

- Calculer le coût de l'algorithme de produit de matrices.
- Calculer le coût de votre algorithme de calcul de $A^*$.
- Qu'en pensez-vous ?

# ANALYSE DU COÛT

- Calculer le coût de l'algorithme de produit de matrices.

# Analyse du coût

- Calculer le coût de l'algorithme de produit de matrices.

> Le produit de matrices correspond à 3 boucles imbriquées de même taille $n$ donc le coût est en $O(n^3)$

- Calculer le coût de votre algorithme de calcul de $A^*$.

# Analyse du coût

- Calculer le coût de l'algorithme de produit de matrices.

Le produit de matrices correspond à 3 boucles imbriquées de même taille $n$ donc le coût est en $O(n^3)$

- Calculer le coût de votre algorithme de calcul de $A^*$.

On itère $n$ fois un produit de matrice, le calcul de la somme étant en $\mathcal{O}(n^2)$ le coût du bloc est $\mathcal{O}(n^3)$ et le coût total en $\mathcal{O}(n^4)$.

- Qu'en pensez-vous ?

# Analyse du coût

- Calculer le coût de l'algorithme de produit de matrices.

Le produit de matrices correspond à 3 boucles imbriquées de même taille $n$ donc le coût est en $O(n^3)$

- Calculer le coût de votre algorithme de calcul de $A^*$.

On itère $n$ fois un produit de matrice, le calcul de la somme étant en $\mathcal{O}(n^2)$ le coût du bloc est $\mathcal{O}(n^3)$ et le coût total en $\mathcal{O}(n^4)$.

- Qu'en pensez-vous ?

Le coût est important, d'autant que les algorithmes de parcours de graphe sont plus efficaces (voir le cours correspondant). Cependant la démarche montre que la décomposition par longueur de chemin conduit à une forme algébrique assez simple à traiter, cela ressemble à une série géométrique

$$1 + a + a^2 + a^3 + \cdots + a^{n-1}$$

qui est un objet mathématique assez simple.

# EXISTENCE D'UN CHEMIN

On rappelle que la matrice booléenne $A^*$ associée à un graphe $\mathcal{G}$ de matrice d'adjacence $A$ représente l'existence des chemins dans $\mathcal{G}$.

$$a^*_{x,y} = \text{il existe un chemin de } x \text{ à } y \text{ dans} \mathcal{G}$$

$A^*$ s'exprime algébriquement avec les opérateurs $\oplus$ et $\otimes$ (dans notre cas *ou* bouléen et *et* booléen).

$$A^* = I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k} \oplus \cdots$$

avec $I$ la matrice diagonale identité (1 sur la diagonale et 0 ailleurs),
Le problème devient donc plus abstrait, complètement algébrique avec des coefficients booléens.

## Questions

- Expliquer en une phrase ce que représente

$$I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k}$$

- Montrer que

$$I \oplus A \oplus A^{\otimes 2} \oplus \cdots \oplus A^{\otimes k} \oplus A^{\otimes (k+1)} = \left(I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k}\right) \otimes (I \oplus A)$$

# EXISTENCE D'UN CHEMIN

- Expliquer en une phrase ce que représente

$$I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k}$$

# Existence d'un chemin

- Expliquer en une phrase ce que représente

$$I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k}$$

$A^{\otimes k}$ est la matrice représentant l'existence de chemins de longueur $k$, le $\oplus$ représentant le *ou* booléen, la somme des matrices représente l'existence d'un chemin de longueur 0 ou de longueur 1 ou de longueur 2 $\cdots$ ou de longueur $k$. C'est à dire l'existence d'un chemin de longueur inférieure ou égale à $k$.

- Montrer que

$$I \oplus A \oplus A^{\otimes 2} \oplus \cdots \oplus A^{\otimes k} \oplus A^{\otimes (k+1)} = \left(I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k}\right) \otimes (I \oplus A)$$

# Existence d'un chemin

- Expliquer en une phrase ce que représente

$$I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k}$$

$A^{\otimes k}$ est la matrice représentant l'existence de chemins de longueur $k$, le $\oplus$ représentant le *ou* booléen, la somme des matrices représente l'existence d'un chemin de longueur 0 ou de longueur 1 ou de longueur 2 $\cdots$ ou de longueur $k$. C'est à dire l'existence d'un chemin de longueur inférieure ou égale à $k$.

- Montrer que

$$I \oplus A \oplus A^{\otimes 2} \oplus \cdots \oplus A^{\otimes k} \oplus A^{\otimes (k+1)} = \left(I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k}\right) \otimes (I \oplus A)$$

La partie gauche de l'égalité représente l'existence de chemin de longueur inférieure à $k+1$. Le premier membre de la partie droite ceux de longueur inférieure à $k$ et le 2$^{\text{ième}}$ membre ceux de longueur inférieure ou égale à 1. Or tout chemin de longueur $\leqslant k+1$ peut se décomposer en un chemin de longueur $\leqslant k$ suivi d'un chemin de longueur $\leqslant 1$ ce qui s'exprime par l'opérateur $\otimes$

# CHEMINS ET PUISSANCE DE MATRICE

## Questions

- En déduire que

$$A^* = (I \oplus A)^{\otimes(n-1)}$$

- En déduire un algorithme de calcul de $A^*$ en $\mathcal{O}(n^3 \log n)$

Le terme en $n^3$ vient de la multiplication des matrices, c'est l'opération coûteuse. On peut la réduire avec des algorithmes comme celui de Strassen mais cela reste difficile

# EXISTENCE D'UN CHEMIN

- En déduire que

$$A^* = (I \oplus A)^{\otimes(n-1)}$$

# EXISTENCE D'UN CHEMIN

- En déduire que

$$A^* = (I \oplus A)^{\otimes(n-1)}$$

L'expression précédente montre que

$$I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k} = (I \oplus A)^{\otimes k}$$

De plus l'existence d'un chemin se réduit à l'existence d'un chemin de longueur $\leqslant n-1$ (cf. précedemment) d'où le résultat.

- En déduire un algorithme de calcul de $A^*$ en $\mathcal{O}(n^3 \log n)$

# EXISTENCE D'UN CHEMIN

- En déduire que

$$A^* = (I \oplus A)^{\otimes(n-1)}$$

L'expression précédente montre que

$$I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \cdots \oplus A^{\otimes k} = (I \oplus A)^{\otimes k}$$

De plus l'existence d'un chemin se réduit à l'existence d'un chemin de longueur $\leqslant n-1$ (cf. précedemment) d'où le résultat.

- En déduire un algorithme de calcul de $A^*$ en $\mathcal{O}(n^3 \log n)$

On a vu de le calcul de $x^n$ pouvait se faire en temps $\mathcal{O}(\log n)$ par la méthode de l'exponentiation rapide, ce qui fonctionne bien ici et donne le résultat.

# Diviser pour régner

Nous allons maintenant montrer que le coût de la fermeture transitive est du même ordre de grandeur que le coût d'une multiplication de matrices. Cette partie est inspirée du livre de Kozen.
Deux points intéressants dans cette partie : l'interprétation de la division du problème en termes de graphe d'une part et le calcul de la complexité d'autre part.
On décompose la matrice d'adjacence en 4 blocs (comme d'habitude on suppose que $n$ est une puissance de 2, on verra après comment ajuster lorsque ce n'est pas le cas)

$$A = \left(\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline A_{2,1} & A_{2,2} \end{array}\right)$$

**Questions**

- Quel sens donner à chacun des 4 blocs $A_{1,1}$, $A_{1,2}$, $A_{2,1}$ et $A_{2,2}$ ?

- Quel sens donner à chacun des 4 blocs $A_{1,1}$, $A_{1,2}$, $A_{2,1}$ et $A_{2,2}$ ?

- Quel sens donner à chacun des 4 blocs $A_{1,1}$, $A_{1,2}$, $A_{2,1}$ et $A_{2,2}$ ?

Cette décomposition de la matrice correspond au partitionnement de l'ensemble des sommets du graphe en 2 , $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ avec $\mathcal{X}_1 = \{x_1, \cdots, x_{\frac{n}{2}}\}$ et $\mathcal{X}_2 = \{x_{\frac{n}{2}+1}, \cdots, x_n\}$.

- $A_{1,1}$ correspond aux arcs de $\mathcal{X}_1$ vers $\mathcal{X}_1$
- $A_{2,2}$ correspond aux arcs de $\mathcal{X}_2$ vers $\mathcal{X}_2$
- $A_{1,2}$ correspond aux arcs de $\mathcal{X}_1$ vers $\mathcal{X}_2$
- $A_{2,1}$ correspond aux arcs de $\mathcal{X}_2$ vers $\mathcal{X}_1$

# DIVISER POUR RÉGNER (2)

L'algorithme se déroule en 5 étapes

Étape 1 : Diviser la matrice $A$ (page précédente)

$$A = \left(\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline A_{2,1} & A_{2,2} \end{array}\right)$$

Étape 2 : Calculer récursivement $A^*_{2,2}$

Étape 3 : Calculer

$$B = A_{1,1} \oplus A_{1,2} \otimes A^*_{2,2} \otimes A_{2,1}$$

Étape 4 : Calculer récursivement $B^*$

Étape 5 : Retourner

$$A^* = \left(\begin{array}{c|c} B^* & B^* \otimes A_{1,2} \otimes A^*_{2,2} \\ \hline A^*_{2,2} \otimes A_{2,1} \otimes B^* & A^*_{2,2} \oplus A^*_{2,2} \otimes A_{2,1} \otimes B^* \otimes A_{1,2} \otimes A^*_{2,2} \end{array}\right)$$

# Analyse de l'algorithme

## Questions

- Que représente la matrice $A^*_{2,2}$ ?
  (un coefficient de cette matrice est vrai ssi il existe ...)
- Que représente la matrice $B = A_{1,1} \oplus A_{1,2} \otimes A^*_{2,2} \otimes A_{2,1}$ ?
- Que représente la matrice $B^*$ ?
- Que représentent chacun des 4 blocs de la matrice renvoyée ?
  - $B^*$ (déjà répondu avant)
  - $B^* \otimes A_{1,2} \otimes A^*_{2,2}$
  - $A^*_{2,2} \otimes A_{2,1} \otimes B^*$
  - $A^*_{2,2} \oplus A^*_{2,2} \otimes A_{2,1} \otimes B^* \otimes A_{1,2} \otimes A^*_{2,2}$

- Que représente la matrice $A^*_{2,2}$ ? (un coefficient de cette matrice est vrai ssi il existe ...)

- Que représente la matrice $A^*_{2,2}$ ? (un coefficient de cette matrice est vrai ssi il existe ...)

> L'existence de chemins restant dans $\mathcal{X}_2$

- Que représente la matrice $B = A_{1,1} \oplus A_{1,2} \otimes A^*_{2,2} \otimes A_{2,1}$ ?

- Que représente la matrice $A^*_{2,2}$ ? (un coefficient de cette matrice est vrai ssi il existe ...)

> L'existence de chemins restant dans $\mathcal{X}_2$

- Que représente la matrice $B = A_{1,1} \oplus A_{1,2} \otimes A^*_{2,2} \otimes A_{2,1}$ ?

> L'existence d'un arc dans $\mathcal{X}_1$ ou l'existence d'un chemin qui part de $\mathcal{X}_1$ vers $\mathcal{X}_2$, reste un certain temps dans $\mathcal{X}_2$ (matrice $A^*_{2,2}$) puis revient par un arc dans $\mathcal{X}_1$

- Que représente la matrice $B^*$ ?

- Que représente la matrice $A^*_{2,2}$ ? (un coefficient de cette matrice est vrai ssi il existe ...)

L'existence de chemins restant dans $\mathcal{X}_2$

- Que représente la matrice $B = A_{1,1} \oplus A_{1,2} \otimes A^*_{2,2} \otimes A_{2,1}$ ?

L'existence d'un arc dans $\mathcal{X}_1$ ou l'existence d'un chemin qui part de $\mathcal{X}_1$ vers $\mathcal{X}_2$, reste un certain temps dans $\mathcal{X}_2$ (matrice $A^*_{2,2}$) puis revient par un arc dans $\mathcal{X}_1$

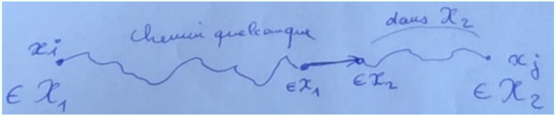
- Que représente la matrice $B^*$ ?

L'existence d'un chemin dans $\mathcal{G}$ d'un sommet de $\mathcal{X}_1$ vers un sommet de $\mathcal{X}_1$

- Que représentent chacun des 4 blocs de la matrice renvoyée ?
  - $B^* \otimes A_{1,2} \otimes A^*_{2,2}$

- Que représentent chacun des 4 blocs de la matrice renvoyée ?
  - $B^* \otimes A_{1,2} \otimes A^*_{2,2}$

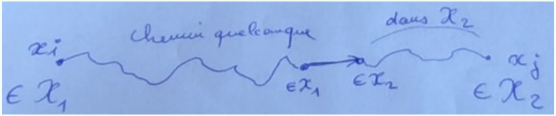
L'existence d'un chemin dans $\mathcal{G}$ d'un sommet de $\mathcal{X}_1$ vers un sommet de $\mathcal{X}_2$.




- $A^*_{2,2} \otimes A_{2,1} \otimes B^*$

- Que représentent chacun des 4 blocs de la matrice renvoyée ?
  - $B^* \otimes A_{1,2} \otimes A^*_{2,2}$

L'existence d'un chemin dans $\mathcal{G}$ d'un sommet de $\mathcal{X}_1$ vers un sommet de $\mathcal{X}_2$.



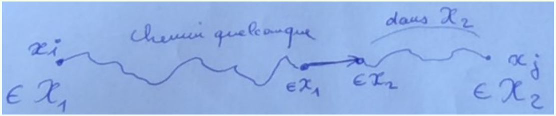

- $A^*_{2,2} \otimes A_{2,1} \otimes B^*$

L'existence d'un chemin dans $\mathcal{G}$ d'un sommet de $\mathcal{X}_2$ vers un sommet de $\mathcal{X}_1$ (faire le dessin)

- $A^*_{2,2} \oplus A^*_{2,2} \otimes A_{2,1} \otimes B^* \otimes A_{1,2} \otimes A^*_{2,2}$

- Que représentent chacun des 4 blocs de la matrice renvoyée ?
  - $B^* \otimes A_{1,2} \otimes A^*_{2,2}$

L'existence d'un chemin dans $\mathcal{G}$ d'un sommet de $\mathcal{X}_1$ vers un sommet de $\mathcal{X}_2$.




- $A^*_{2,2} \otimes A_{2,1} \otimes B^*$

L'existence d'un chemin dans $\mathcal{G}$ d'un sommet de $\mathcal{X}_2$ vers un sommet de $\mathcal{X}_1$ (faire le dessin)

- $A^*_{2,2} \oplus A^*_{2,2} \otimes A_{2,1} \otimes B^* \otimes A_{1,2} \otimes A^*_{2,2}$

L'existence d'un chemin dans $\mathcal{G}$ d'un sommet de $\mathcal{X}_2$ vers un sommet de $\mathcal{X}_2$ (faire le dessin)

# ANALYSE DU COÛT DE L'ALGORITHME (PARTIE DIFFICILE)

On suppose que le coût d'une multiplication $\otimes$ de matrices de dimension $n$ est $m(n)$ (avec le produit usuel on a $\mathcal{O}(n^3)$), le coût d'une somme $\otimes$ est clairement en $\mathcal{O}(n^2)$ car il y a $n^2$ coefficients à calculer. On note $T(n)$ le coût de l'algorithme.

## Questions

- Calculer le nombre d'opérations $\otimes$ et $\oplus$ effectuées par l'algorithme.
- Remarquer que des calculs sont répétés, par exemple $A_{1,2} \otimes A^*_{2,2}$ est calculé à l'étape 3 et peut être réutilisé à l'étape 5 pour le coefficient ligne 1 colonne 2. Quelles autres réutilisations des calculs peut-on faire ?
- Écrire l'équation de complexité vérifiée par $T(n)$.
- Résoudre cette équation on pourra supposer que $m(n) \geqslant 4m(\frac{m}{2})$.

- Calculer le nombre d'opérations $\otimes$ et $\oplus$ effectuées par l'algorithme.

- Calculer le nombre d'opérations $\otimes$ et $\oplus$ effectuées par l'algorithme.

2 opérations $\oplus$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. Le coût de $\oplus$ est de l'ordre de $\left(\frac{n}{2}\right)^2$

10 opérations $\times$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. le coût de $\otimes$ est de l'ordre de $m(\frac{n}{2})$

- Remarquer que des calculs sont répétés, par exemple $A_{1,2} \otimes A^*_{2,2}$ est calculé à l'étape 3 et peut être réutilisé à l'étape 5 pour le coefficient ligne 1 colonne 2. Quelles autres réutilisations des calculs peut-on faire ?

- Calculer le nombre d'opérations $\otimes$ et $\oplus$ effectuées par l'algorithme.

> 2 opérations $\oplus$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. Le coût de $\oplus$ est de l'ordre de $\left(\frac{n}{2}\right)^2$
>
> 10 opérations $\times$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. le coût de $\otimes$ est de l'ordre de $m(\frac{n}{2})$

- Remarquer que des calculs sont répétés, par exemple $A_{1,2} \otimes A^*_{2,2}$ est calculé à l'étape 3 et peut être réutilisé à l'étape 5 pour le coefficient ligne 1 colonne 2. Quelles autres réutilisations des calculs peut-on faire ?

> $A^*_{2,2} \otimes A_{2,1} \otimes B^*$ et $B^* \otimes A_{1,2} \otimes A^*_{2,2}$ dans le dernier terme. Ce qui fait 6 $\otimes$

- Écrire l'équation de complexité vérifiée par $T(n)$.

- Calculer le nombre d'opérations $\otimes$ et $\oplus$ effectuées par l'algorithme.

2 opérations $\oplus$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. Le coût de $\oplus$ est de l'ordre de $\left(\frac{n}{2}\right)^2$

10 opérations $\times$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. le coût de $\otimes$ est de l'ordre de $m(\frac{n}{2})$

- Remarquer que des calculs sont répétés, par exemple $A_{1,2} \otimes A^*_{2,2}$ est calculé à l'étape 3 et peut être réutilisé à l'étape 5 pour le coefficient ligne 1 colonne 2. Quelles autres réutilisations des calculs peut-on faire ?

$A^*_{2,2} \otimes A_{2,1} \otimes B^*$ et $B^* \otimes A_{1,2} \otimes A^*_{2,2}$ dans le dernier terme. Ce qui fait 6 $\otimes$

- Écrire l'équation de complexité vérifiée par $T(n)$.

$$T(n) = 2T(\frac{n}{2}) + cm(\frac{n}{2}) + d(\frac{n}{2}) \text{ avec } c \text{ et } d \text{ des coefficients constants}$$

- Résoudre cette équation on pourra supposer que $m(n) \geqslant 4m(\frac{m}{2})$.

- Calculer le nombre d'opérations $\otimes$ et $\oplus$ effectuées par l'algorithme.

2 opérations $\oplus$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. Le coût de $\oplus$ est de l'ordre de $\left(\frac{n}{2}\right)^2$

10 opérations $\times$ sur des matrices de taille $\frac{n}{2}$, et celles dans les appels récursifs. le coût de $\otimes$ est de l'ordre de $m(\frac{n}{2})$

- Remarquer que des calculs sont répétés, par exemple $A_{1,2} \otimes A^*_{2,2}$ est calculé à l'étape 3 et peut être réutilisé à l'étape 5 pour le coefficient ligne 1 colonne 2. Quelles autres réutilisations des calculs peut-on faire ?

$A^*_{2,2} \otimes A_{2,1} \otimes B^*$ et $B^* \otimes A_{1,2} \otimes A^*_{2,2}$ dans le dernier terme. Ce qui fait 6 $\otimes$

- Écrire l'équation de complexité vérifiée par $T(n)$.

$$T(n) = 2T(\frac{n}{2}) + cm(\frac{n}{2}) + d(\frac{n}{2}) \text{ avec } c \text{ et } d \text{ des coefficients constants}$$

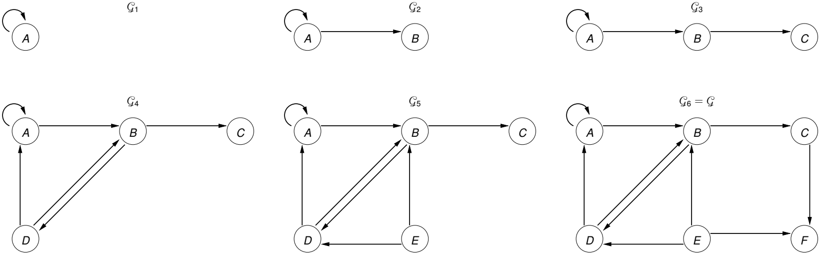
- Résoudre cette équation on pourra supposer que $m(n) \geqslant 4m(\frac{m}{2})$.

on trouve $T(n) = \mathcal{O}(m(n))$

UGA Université Grenoble Alpes

# ITÉRATION SUR LE NOMBRE DE SOMMETS

Dans le calcul de $A^*$ matrice de l'existence d'un chemin, on a décomposé le problème par rapport à la longueur des chemins (matrice $A^{\otimes k}$ pour les chemins de longueur $k$.
Dans cette partie on va décomposer par rapport à l'ensemble des sommets, tout se passe comme si on introduit les sommets un par un et que l'on calcule la fermeture transitive au fur et à mesure.
On note $\mathcal{G}_1, \mathcal{G}_2, \cdots \mathcal{G}_n = \mathcal{G}$ les sous-graphes successifs de $\mathcal{G}$, la matrice d'adjacence $A_k$ de $\mathcal{G}_k$ est juste la sous-matrice de $A$ constituée des $k$ premières lignes et des $k$ premières colonnes.



# Itération sur le nombre de sommets(2)

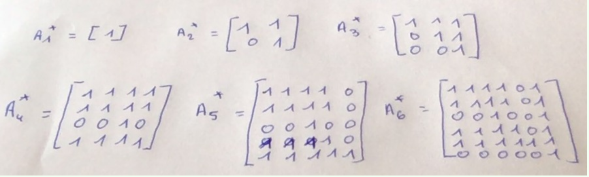
## Questions

Travail exploratoire de recherche, on va exprimer $A_k^*$ en fonction de $A_{k-1}^*$ et d'éléments de la $k^{\text{ième}}$ ligne et de la $k^{\text{ième}}$ colonne d'indice $A$.

- Regarder ce qui se passe sur l'exemple.
- Comment calculer les éléments de la $k^{\text{ième}}$ colonne de $A_k^*$ ?
  Donner une interprétation en termes de chemins.
- Comment calculer les éléments de la $k^{\text{ième}}$ ligne de $A_k^*$ ?
  Donner une interprétation en termes de chemins.
- Comment calculer le reste des éléments de $A_k^*$ ?
  Donner une interprétation en termes de chemins.
- Écrire un algorithme qui calcule $A^*$.
- Évaluer le coût de cet algorithme.

- Regarder ce qui se passe sur l'exemple.

- Regarder ce qui se passe sur l'exemple.



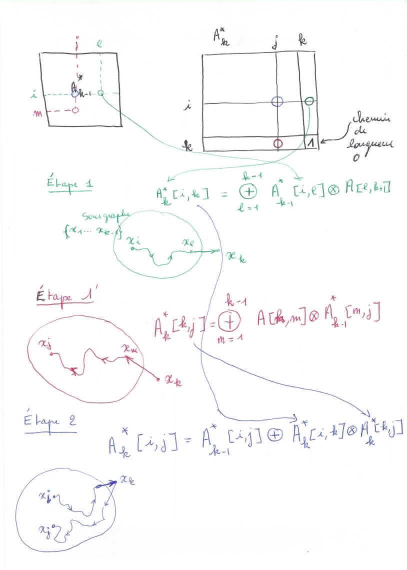
On remarque que la matrice $A^*_{k+1}$ contient la matrice $A^*_k$ ou presque (il peut y avoir des modif mais grossièrement la matrice grandit). D'ailleurs on ne pourrait n'utiliser qu'une seule matrice pour faire le calcul

- Comment calculer les éléments de la $k^{\text{ième}}$ colonne de $A_k^*$ ?
  Donner une interprétation en termes de chemins.
- Comment calculer les éléments de la $k^{\text{ième}}$ ligne de $A_k^*$ ?
  Donner une interprétation en termes de chemins.
- Comment calculer le reste des éléments de $A_k^*$ ?
  Donner une interprétation en termes de chemins.

On peut tout replier sur une même matrice, ... cela ne fait-il pas penser à de la programmation dynamique ?

- Comment calculer les éléments de la $k^{\text{ième}}$ colonne de $A_k^*$ ? Donner une interprétation en termes de chemins.
- Comment calculer les éléments de la $k^{\text{ième}}$ ligne de $A_k^*$ ? Donner une interprétation en termes de chemins.
- Comment calculer le reste des éléments de $A_k^*$ ? Donner une interprétation en termes de chemins.

On peut tout replier sur une même matrice, ... cela ne fait-il pas penser à de la programmation dynamique ?



```
CheminD (A)
    Données: A matrice d'adjacence n × n
    Résultat: matrice A*

    A* matrice n × n initialisée à 0
    A*_{1,1} = 1
    for k = 2 to n // calcul de A*_k
        for i = 1 to k − 1 // traitement de la colonne k
            for l = 1 to k − 1
                A*_{i,k} = A*_{i,k} ⊕ A*_{i,l} ⊗ A_{l,k}
        for j = 1 to k − 1 // traitement de la ligne k
            for m = 1 to k − 1
                A*_{k,j} = A*_{k,j} ⊕ A_{k,m} ⊗ A*_{m,j}
        foreach (i, j) ∈ {1, ··· , k − 1}^2 do // bloc de taille (k − 1) × (k − 1)
            A*_{i,j} = A*_{i,j} ⊕ A*_{i,k} ⊗ A*_{k,j}
        A*_{k,k} = 1
    Return A*
```

Chaque boucle interne effectue $(k-1)^2$ opérations $\oplus$ et $\otimes$, au total on aura donc de l'ordre de $\mathcal{O}(n^3)$ opérations.

Rappel : $1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6} = \mathcal{O}(n^3)$

# SYNTHÈSE

La représentation matricielle du graphe et les opérations algébriques que l'on peut réaliser sur les lignes et les colonnes montre que le calculs de l'existence de chemins se résume au calcul d'une matrice $A^*$.
La complexité des algorithmes observés jusqu'à maintenant sont de l'ordre de $\mathcal{O}(n^3)$ ($\mathcal{O}(n^3 \log n)$ avec la puissance).

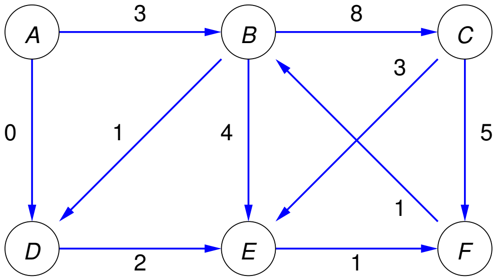
## Chemins de coût minimal

On considère maintenant que la matrice d'adjacence contient le coût des arcs

$$a_{i,j} = \begin{cases} \text{coût de l'arc } (x_i, x_j) & \text{s'il existe un arc de } x_i \text{ à } x_j \\ +\infty & \text{sinon} \end{cases}$$

On suppose dans un premier temps que les coûts sont positifs.
L'objectif est de calculer la matrice $M$ dont le coefficient $m_{i,j}$ est le coût minimum d'un chemins de $x_i$ à $x_j$ ($+\infty$ s'il n'existe pas un tel chemin).

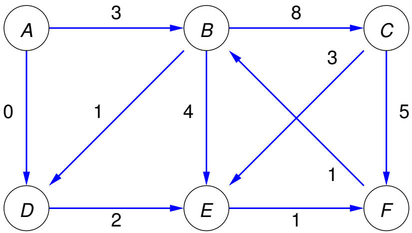
# EXEMPLE




**Questions**

- Écrire la matrice des coûts associée à ce graphe
- Écrire la matrice des coûts des plus courts chemins de longueur 2 associée à ce graphe

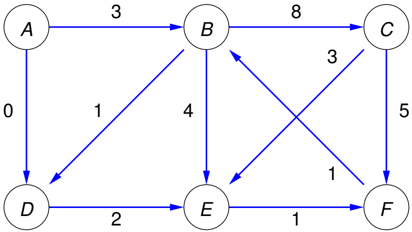
# Exemple




## Questions

- Écrire la matrice des coûts associée à ce graphe
- Écrire la matrice des coûts des plus courts chemins de longueur 2 associée à ce graphe

# Exemple




**Questions**

- Écrire la matrice des coûts associée à ce graphe
- Écrire la matrice des coûts des plus courts chemins de longueur 2 associée à ce graphe

(notation $\infty$ signifie $+\infty$)

$$A = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & \infty & 3 & \infty & 0 & \infty & \infty \\ B & \infty & \infty & 8 & 1 & 4 & \infty \\ C & \infty & \infty & \infty & \infty & 3 & 5 \\ D & \infty & \infty & \infty & \infty & 2 & \infty \\ E & \infty & \infty & \infty & \infty & \infty & 1 \\ F & \infty & 1 & \infty & \infty & \infty & \infty \end{array} \qquad A^{\otimes 2} = \begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & \infty & \infty & 11 & 4 & 7 & \infty \\ B & \infty & \infty & \infty & \infty & 3 & 5 \\ C & \infty & 6 & \infty & \infty & \infty & 4 \\ D & \infty & \infty & \infty & \infty & \infty & 3 \\ E & \infty & 2 & \infty & \infty & \infty & \infty \\ F & \infty & \infty & 9 & 2 & 5 & \infty \end{array}$$

# EXEMPLE

C'est évident, on a fait le même travail que celui fait pour obtenir l'existence d'un chemin sauf que l'on a remplacé le *ou* par $\min$ et le *et* par $+$. les opérateurs ont été remplacés, la matrice a une autre interprétation et on a exactement les mêmes algorithmes.

**Questions**

- Réécrire l'algorithme du calcul des chemins de coût minimum en transformant `CheminD` de manière adéquate (c'est l'algorithme de Dantzig de calcul des chemins de coût minimal).
- Exécuter cet algorithme sur l'exemple
- Augmenter cet algorithme de manière à obtenir la matrice $P$ les prédécesseurs des chemins sur le graphe et donner la fonction qui calcule un chemin de coût minimal

- Réécrire l'algorithme du calcul des chemins de coût minimum en transformant `CheminD` de manière adéquate (c'est l'algorithme de Dantzig de calcul des chemins de coût minimal).

- Augmenter cet algorithme de manière à obtenir la matrice $P$ les prédécesseurs des chemins sur le graphe et donner la fonction qui calcule un chemin de coût minimal

```
Algorithme_de_Dantzig (A)
  Données: A matrice d'adjacence des coûts n × n
  Résultat: matrice des coûts minimum A*, la matrice P, P_{i,j} = prédécesseur de j sur le chemin x_i ⇝ x_j

  A* matrice n × n initialisée à +∞
  A*_{1,1} = 0
  for k = 2 to n // calcul de A*_k
      for i = 1 to k − 1 // traitement de la colonne k
          for l = 1 to k − 1
              if A*_{i,l} + A_{l,k} ⩽ A*_{i,k}
                  A*_{i,k} = A*_{i,l} + A_{l,k}
                  P_{i,k} = l
      for j = 1 to k − 1 // traitement de la ligne k
          for m = 1 to k − 1
              if A_{k,m} + A*_{m,j} ⩽ A*_{k,j}
                  A*_{k,j} = A_{k,m} + A*_{m,j} ⩽ A*_{k,j}
                  P_{k,j} = P_{m,j}
      foreach (i, j) ∈ {1, · · · , k − 1}^2 do // bloc de taille (k − 1) × (k − 1)
          if A*_{i,k} + A*_{k,j} ⩽ A*_{i,j}
              A*_{i,j} = A*_{i,k} + A*_{k,j}
              P_{i,j} = P_{k,j}
      A*_{k,k} = 0
  Return A*, P
```

# Encore plus fort

On suppose maintenant que les coefficients de la matrice $A$ sont des ensembles de chemins de longueur 1. On remplace l'opérateur $\oplus$ par l'union des ensembles et $\otimes$ par la concaténation des chemins.

## Questions

- Que contient la matrice $(I \oplus A)^{\otimes k}$ ? ($I$ est la matrice des chemins de longueur 0)
- Comment interpréter le travail effectué précédemment ?

On suppose maintenant que les coefficients de la matrice $A$ sont des ensembles de chemins de longueur 1. On remplace l'opérateur $\oplus$ par l'union des ensembles et $\otimes$ par la concaténation des chemins (produit cartésien).

- Que contient la matrice $(I \oplus A)^{\otimes k}$ ? ($I$ est la matrice des chemins de longueur 0)
- Comment interpréter le travail effectué précédemment ?

On suppose maintenant que les coefficients de la matrice $A$ sont des ensembles de chemins de longueur 1. On remplace l'opérateur $\oplus$ par l'union des ensembles et $\otimes$ par la concaténation des chemins (produit cartésien).

- Que contient la matrice $(I \oplus A)^{\otimes k}$ ? ($I$ est la matrice des chemins de longueur 0)
- Comment interpréter le travail effectué précédemment ?

$(I \oplus A)^{\otimes k}_{i,j}$ est un ensemble qui contient tous les chemins $x_i \rightsquigarrow x_j$ de longueur (en nombre d'arcs) inférieure ou égale à $k$
Donc $(I \oplus A)^{*}_{i,j}$ est un ensemble qui contient tous les chemins $x_i \rightsquigarrow x_j$ dans $\mathcal{G}$.
Attention cet ensemble est potentiellement infini, mais ce n'est pas grave, on ne va pas coder tout cet ensemble de chemins.

La première partie du cours consiste simplement à tester si cet ensemble est vide ou pas, il est évident que l'on a un schéma croissant, dès que l'on a un chemin l'ensemble est non vide et ce n'est pas la peine d'aller plus loin.

Dans la deuxième partie, on associe à chaque chemin un coût et on cherche le coût minimum, comme on suppose les coûts des arcs positifs l'existence d'un chemin implique l'existence d'un chemin de coût minimum (qui est élémentaire). Là encore il n'est pas nécessaire de parcourir tous les chemins.