

# Algorithmique des graphes - Cours 6

Olivier Baudon

Université de Bordeaux

11 octobre 2021

## Objectifs

L'algorithme de Bellman-Ford, que nous appellerons simplement **Ford** (pour éviter les confusions), permet de calculer les plus courts chemins entre un sommet  $s$  et tous les autres, en particulier même si le graphe possède un ou des circuits et/ou des arcs de poids négatifs.

Si le graphe possède un circuit négatif, l'algorithme renverra Faux, sinon Vrai.

## Principe

Le principe de l'algorithme est basé sur la constatation suivante : s'il existe un plus court chemin entre deux sommets  $u$  et  $v$  dans un graphe à  $n$  sommets, il existe un plus court chemin de longueur au plus  $n - 1$ .

On va donc calculer les plus courts chemins depuis un sommet  $s$  ayant successivement des longueurs (au moins de) 1 à  $n - 1$ . Puis on va calculer les plus courts chemins depuis un sommet  $s$  ayant une longueur (au moins)  $n$ . Si la longueur des plus courts chemins de longueur (au moins)  $n$  diminue par rapport à ceux de longueur (au moins)  $n - 1$ , alors on pourra en déduire que le graphe possède un circuit négatif.

# Algorithme de Ford

## Principe

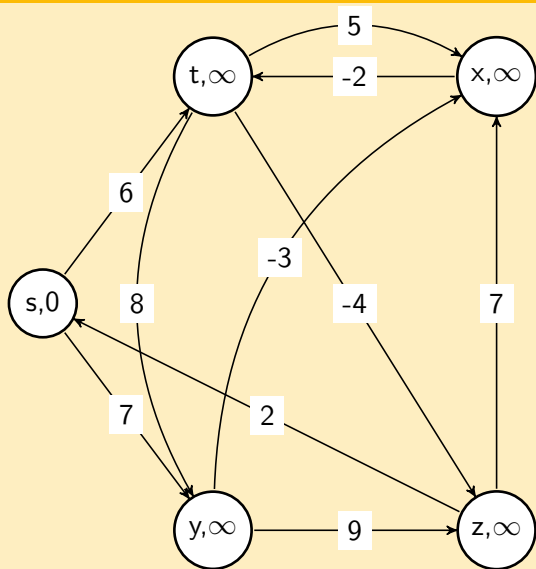
A chaque itération de 1 à  $n$ , on effectue un relâchement sur chaque arc  $uv$  pour voir si il est opportun d'utiliser cet arc pour avoir un plus court chemin entre  $s$  et  $v$ .

## Énoncé

Voir document "Algorithmes des graphes" page 8.

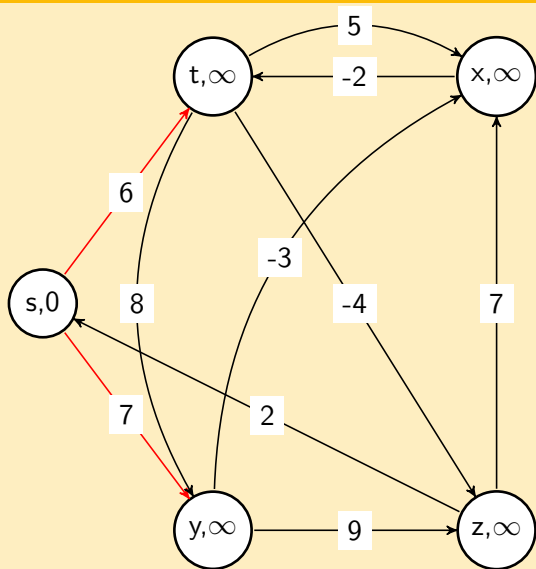
# Algorithme de Ford

## Exemple



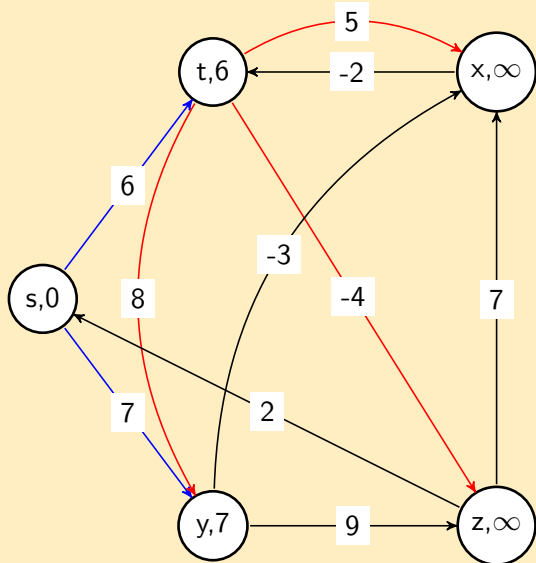
# Algorithme de Ford

Exemple :  $i = 1$ , arcs sortants de  $s$



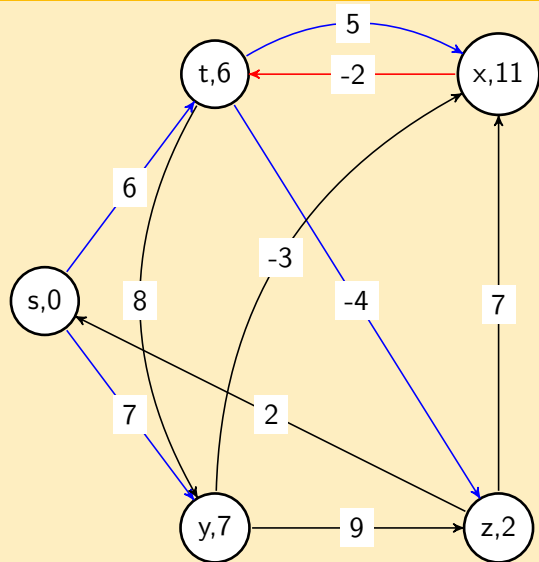
# Algorithme de Ford

Exemple :  $i = 1$ , arcs sortants de  $t$



# Algorithme de Ford

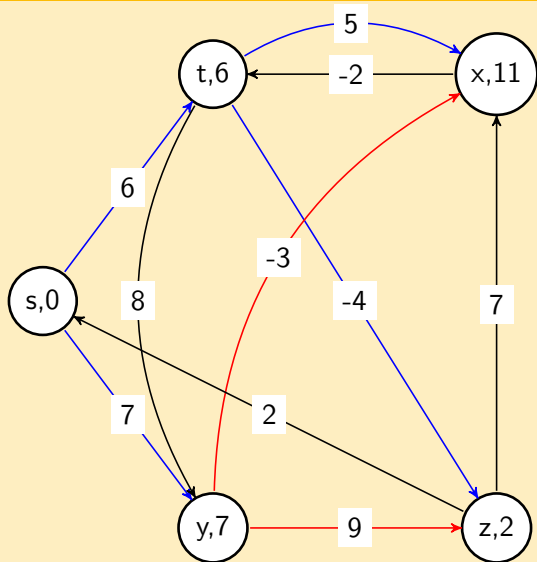
Exemple :  $i = 1$ , arcs sortants de  $x$





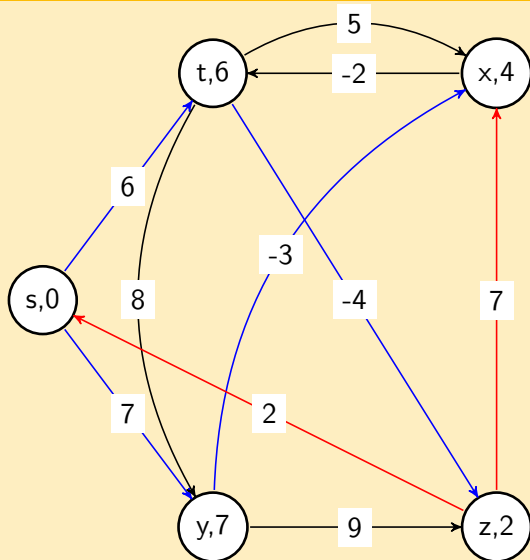
# Algorithme de Ford

Exemple :  $i = 1$ , arcs sortants de  $y$



# Algorithme de Ford

Exemple :  $i = 1$ , arcs sortants de  $z$

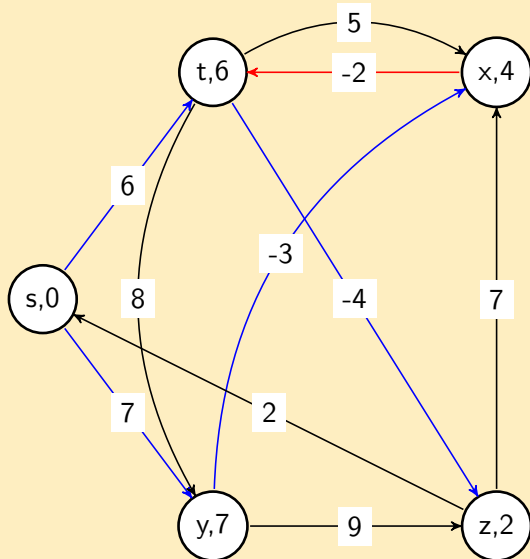


Exemple :  $i = 2$

Remarque : pour  $s$  et  $t$ , leur distance à  $s$  n'a pas été modifiée après avoir traité leurs arcs sortants avec  $i = 1$ .

# Algorithme de Ford

Exemple :  $i = 2$ , arcs sortants de  $x$



Exemple :  $i = 2$

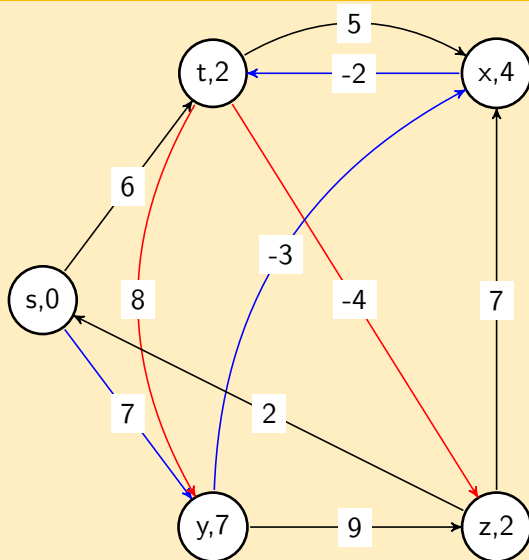
Remarque : pour  $y$  et  $z$ , leur distance à  $s$  n'a pas été modifiée après avoir traité leurs arcs sortants avec  $i = 1$ .

Exemple :  $i = 3$

Remarque : pour  $s, x, y$  et  $z$ , leur distance à  $s$  n'a pas été modifiée après avoir traité leurs arcs sortants avec  $i = 2$ .

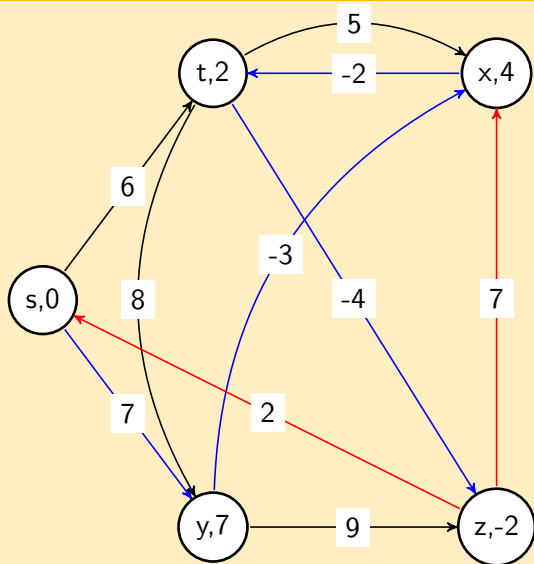
# Algorithme de Ford

Exemple :  $i = 3$ , arcs sortants de  $t$



# Algorithme de Ford

Exemple :  $i = 3$ , arcs sortants de  $z$



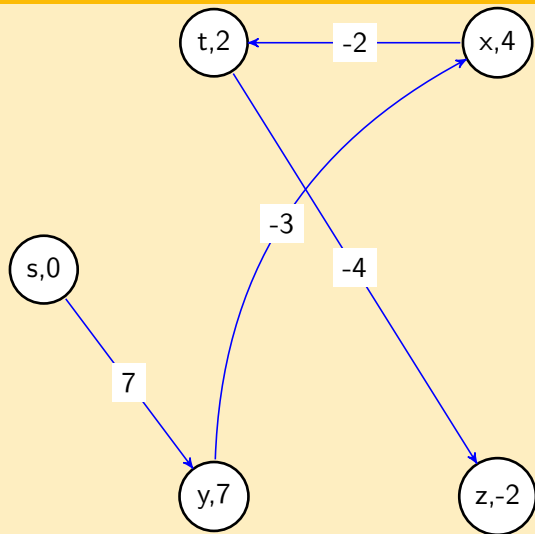


Exemple :  $i = 4$

Remarque : Aucun sommet n'a eu sa distance à  $s$  modifiée après avoir traité ses arcs sortants avec  $i = 3$ . L'algorithme va donc terminer en renvoyant Vrai.

# Algorithme de Ford

## Arborescence



## Complexité

L'algorithme de Ford est en  $O(n \times m)$ .

# Plus court chemin entre toutes paires de sommets

## Utilisation de Dijkstra

Si le graphe n'a que des poids positifs, on peut utiliser  $n$  fois Dijkstra (1 fois à partir de chaque sommet). On obtient ainsi un algorithme en  $O(n^2 \times \log(n) + n \times m)$  en utilisant un tas de Fibonacci pour coder la file de priorité.

# Plus court chemin entre toutes paires de sommets

## Calcul matriciel

On considère une matrice  $W$  avec  $W[i, j] =$

- ▶ 0 si  $i = j$
- ▶  $\omega(i, j)$  si  $ij \in E(G)$
- ▶  $\infty$  sinon.

# Plus court chemin entre toutes paires de sommets

## Calcul matriciel

On calcule ensuite une suite de matrices  $D^{(1)} \dots D^{(n-1)}$ . On pose  $D^{(1)} = W$ . La valeur  $D_{ij}^{(k)}$  contiendra la longueur d'un plus court chemin de  $i$  à  $j$  de longueur au moins  $k$ .  $D^{(k+1)}$  est calculée à partir de  $D^{(k)}$  à l'aide de l'algorithme suivant :

- 1: **pour**  $i$  de 1 à  $n$  **faire**
- 2:     **pour**  $j$  de 1 à  $n$  **faire**
- 3:          $D_{ij}^{(k+1)} \leftarrow \infty$
- 4:         **pour**  $a$  de 1 à  $n$  **faire**
- 5:              $D_{ij}^{(k+1)} \leftarrow \min(D_{ij}^{(k+1)}, D_{ia}^{(k)} + W_{aj})$
- 6:         **fin pour**
- 7:     **fin pour**
- 8: **fin pour**
- 9: **retourner**  $D^{(k+1)}$

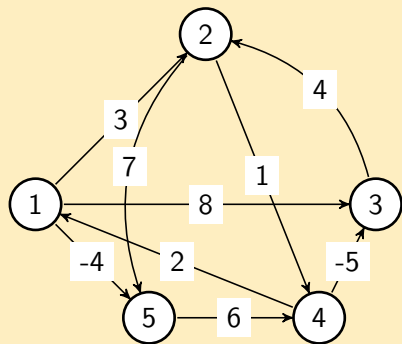
# Plus court chemin entre toutes paires de sommets

## Calcul matriciel

Si  $D^{(n-1)}$  contient dans sa diagonale une valeur négative, disons en  $D_{ii}^{(n-1)}$ , alors le graphe possède un circuit négatif passant par le sommet d'indice  $i$ .

# Plus court chemin entre toutes paires de sommets

## Calcul matriciel - Exemple



Grappe  $G$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

Matrice  $W$



# Plus court chemin entre toutes paires de sommets

## Calcul matriciel - Exemple

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(1)} = W$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

$$D^{(2)}$$

# Plus court chemin entre toutes paires de sommets

## Calcul matriciel - Exemple

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

$D^{(2)}$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$W$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$D^{(3)}$

# Plus court chemin entre toutes paires de sommets

## Calcul matriciel - Exemple

$$1 \begin{pmatrix} 0 & 3 & -3 & 2 & -4 \\ 2 & 3 & 0 & -4 & -1 \\ 3 & 7 & 4 & 0 & 11 \\ 4 & 2 & -1 & -5 & 0 \\ 5 & 8 & 5 & 1 & 6 \end{pmatrix}$$

$D^{(3)}$

$$1 \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ 2 & \infty & 0 & \infty & 1 \\ 3 & \infty & 4 & 0 & \infty \\ 4 & 2 & \infty & -5 & 0 \\ 5 & \infty & \infty & \infty & 6 \end{pmatrix}$$

$W$

$$1 \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 2 & 3 & 0 & -4 & -1 \\ 3 & 7 & 4 & 0 & 3 \\ 4 & 2 & -1 & -5 & 0 \\ 5 & 8 & 5 & 1 & 6 \end{pmatrix}$$

$D^{(4)}$

# Plus court chemin entre toutes paires de sommets

## Calcul matriciel - complexité

La complexité du calcul de  $D^{(n-1)}$  est en  $O(n^4)$ .

## Calcul matriciel - accélération

On peut diminuer le nombre de calculs en calculant directement, à partir de  $D^{(k)}$  la matrice  $D^{(2^k)}$  en remplaçant  $W$  par  $D^{(k)}$ . On ne calculera ainsi que les matrices  $D^{(k)}$  avec  $k$  puissance de 2, jusqu'à avoir  $k \geq n - 1$ .

La complexité de la version accélérée est donc en  $O(n^3 \times \log(n))$ .

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Principe

L'algorithme de Floyd (souvent appelé Floyd-Warshall) s'appuie sur le fait que l'on ne va à chaque calcul de la nouvelle matrice considérer que les  $k$  premiers sommets comme sommets intermédiaires potentiels. Comme il y a  $n$  boucles, on aura bien à la fin considéré tous les sommets comme sommet intermédiaire potentiel.

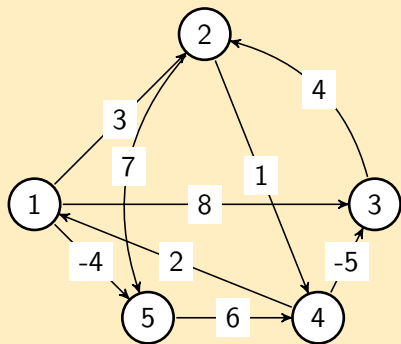
On calculera en parallèle une matrice  $Pred$  où  $Pred_{ij}$  sera l'indice du sommet précédent  $j$  dans le plus court chemin de  $i$  à  $j$ .

## Algorithme de Floyd - Énoncé

Voir document "Algorithmes des graphes" page 6.

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Exemple



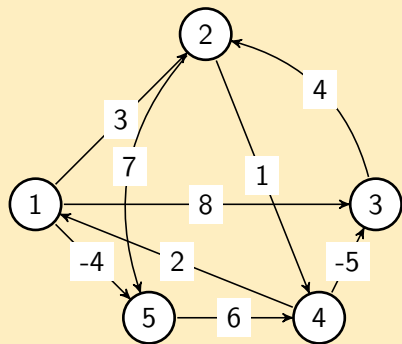
Graphe  $G$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

Matrice  $W$

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Exemple



Graphe G

1	$\left( \begin{array}{ccccc} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & \text{NIL} \\ 4 & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{array} \right)$
2	
3	
4	
5	

Matrice Pred

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Exemple avec $k = 1$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

*Matrices W*

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & NIL & NIL \\ 4 & NIL & 4 & NIL & NIL \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & NIL & NIL \\ 4 & 1 & 4 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

*Matrices Pred*



# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Exemple avec $k = 2$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

*Matrices W*

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & NIL & NIL \\ 4 & 1 & 4 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 1 & 2 & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 2 & 2 \\ 4 & 1 & 4 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

*Matrices Pred*

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Exemple avec $k = 3$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

*Matrices W*

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 1 & 2 & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 2 & 2 \\ 4 & 1 & 4 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 1 & 2 & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 2 & 2 \\ 4 & 3 & 4 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

*Matrices Pred*

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Exemple avec $k = 4$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

*Matrices W*

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 1 & 2 & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 2 & 2 \\ 4 & 3 & 4 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{pmatrix} NIL & 1 & 4 & 2 & 1 \\ 4 & NIL & 4 & 2 & 1 \\ 4 & 3 & NIL & 2 & 1 \\ 4 & 3 & 4 & NIL & 1 \\ 4 & 3 & 4 & 5 & NIL \end{pmatrix}$$

*Matrices Pred*

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Exemple avec $k = 5$

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{pmatrix} \text{NIL} & 1 & 4 & 2 & 1 \\ 4 & \text{NIL} & 4 & 2 & 1 \\ 4 & 3 & \text{NIL} & 2 & 1 \\ 4 & 3 & 4 & \text{NIL} & 1 \\ 4 & 3 & 4 & 5 & \text{NIL} \end{pmatrix}$$

*Matrices W*

*Matrices Pred*

# Plus court chemin entre toutes paires de sommets

## Algorithme de Floyd - Complexité

La complexité de l'algorithme de Floyd est en  $O(n^3)$ .