

Cours de Graphes & Applications (G&A)

Chapitre 2 : Recherche du plus court chemin

Nehla DEBBABI

Année Universitaire : 2022-2023

30 septembre 2023



Plan

1 Introduction

- Motivation
- Définitions & Propriétés

2 Algorithme de DIJKSTRA

3 Algorithme de FORD BELLMAN

4 Algorithme de FLOYD-WARSHALL

5 DIJKSTRA / FORD-BELLMAN / FLOYD-WARSHALL



Google's map : le plus rapide / court chemin vers la destination

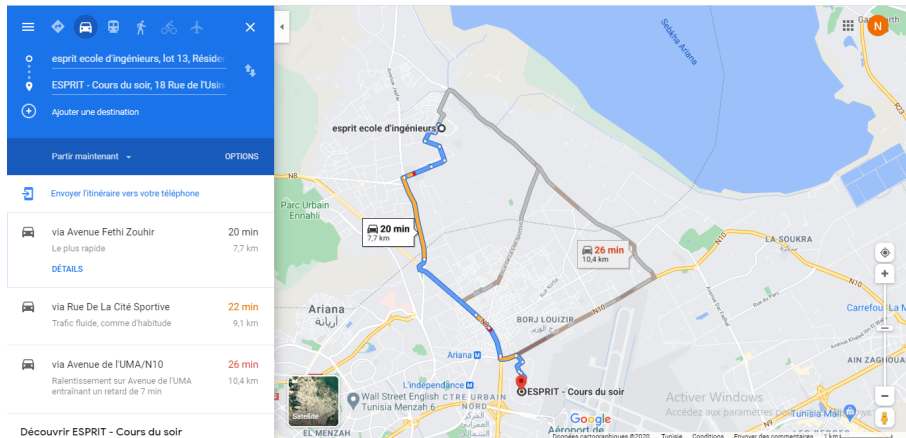


Figure – Itinéraire de ESPRIT Ghazela à ESPRIT Charguia

Le routage : acheminement des paquets par le plus court chemin vers la destination

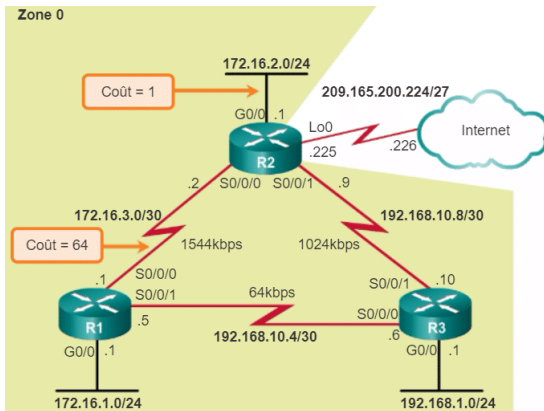


Figure – Illustration d'un réseaux de communication (source de l'image : internet)



Poids / longueur d'un chemin

Définitions

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\mathcal{C}_{i,j}$ **le poids** ou encore **le coût** associer à l'arc $a = (x_i, x_j) \in A$ avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$.

Le poids $\mathcal{C}_{i,j}$ est donné par :

$$\mathcal{C}_{i,j} = \mathcal{C}(x_i, x_j) = \mathcal{C}(a).$$



Poids / longueur d'un chemin

Définitions

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\mathcal{C}_{i,j}$ **le poids** ou encore **le coût** associer à l'arc $a = (x_i, x_j) \in A$ avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$.

Le poids $\mathcal{C}_{i,j}$ est donné par :

$$\mathcal{C}_{i,j} = \mathcal{C}(x_i, x_j) = \mathcal{C}(a).$$

Remarque : les poids peuvent aussi être négatifs.

Poids / longueur d'un chemin

Définitions

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\mathcal{C}_{i,j}$ **le poids** ou encore **le coût** associer à l'arc $a = (x_i, x_j) \in A$ avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$.

Le poids $\mathcal{C}_{i,j}$ est donné par :

$$\mathcal{C}_{i,j} = \mathcal{C}(x_i, x_j) = \mathcal{C}(a).$$

Remarque : les poids peuvent aussi être négatifs.

- Soit $p = \langle s_1, s_2, \dots, s_k \rangle$ un chemin reliant k sommet de $G : s_i \in X, 1 \leq i \leq k$. La **longueur** du chemin p , notée $l(p)$, correspond à la somme des poids associés aux arcs qui le composent :

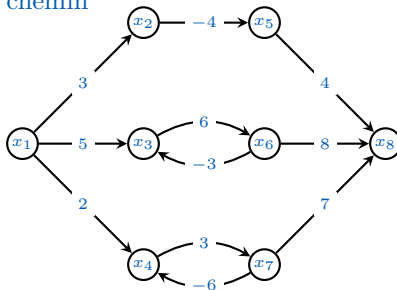
$$l(p) = \sum_{i=1}^{k-1} \mathcal{C}(s_i, s_{i+1})$$



Définitions & Propriétés

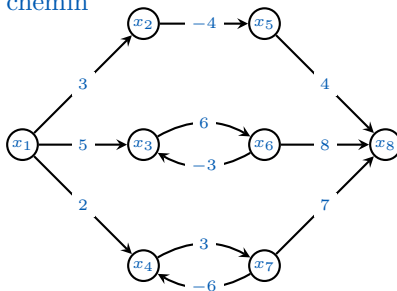
Poids / longueur d'un chemin

Exemple :



Poids / longueur d'un chemin

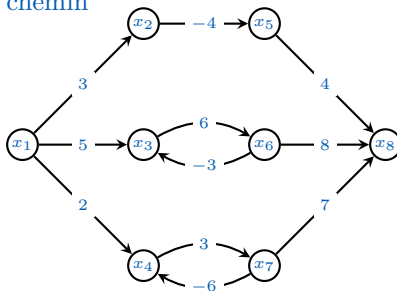
Exemple :



- La longueur du chemin $p_1 = \langle x_1, x_2, x_5, x_8 \rangle$ est $l(p_1)$ avec $l(p_1) = c_{1,2} + c_{2,5} + c_{5,8} = 3 - 4 + 4 = 3$.

Poids / longueur d'un chemin

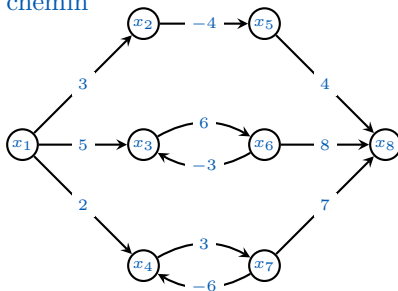
Exemple :



- La longueur du chemin $p_1 = \langle x_1, x_2, x_5, x_8 \rangle$ est $l(p_1)$ avec $l(p_1) = \mathcal{C}_{1,2} + \mathcal{C}_{2,5} + \mathcal{C}_{5,8} = 3 - 4 + 4 = 3$.
- La longueur du chemin $p_2 = \langle x_1, x_3, x_6, x_8 \rangle$ est $l(p_2) = \mathcal{C}_{1,3} + \mathcal{C}_{3,6} + \mathcal{C}_{6,8} = 19$.

Poids / longueur d'un chemin

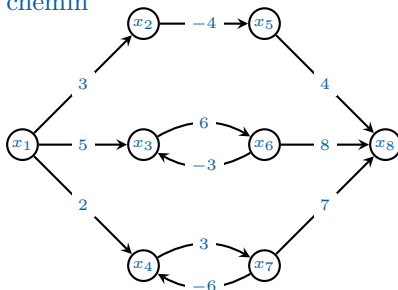
Exemple :



- La longueur du chemin $p_1 = \langle x_1, x_2, x_5, x_8 \rangle$ est $l(p_1)$ avec $l(p_1) = \mathcal{C}_{1,2} + \mathcal{C}_{2,5} + \mathcal{C}_{5,8} = 3 - 4 + 4 = 3$.
- La longueur du chemin $p_2 = \langle x_1, x_3, x_6, x_8 \rangle$ est $l(p_2) = \mathcal{C}_{1,3} + \mathcal{C}_{3,6} + \mathcal{C}_{6,8} = 19$.
- La longueur du chemin $p_3 = \langle x_1, x_3, x_6, x_3, x_6, x_8 \rangle$ est $l(p_3) = 22$.

Poids / longueur d'un chemin

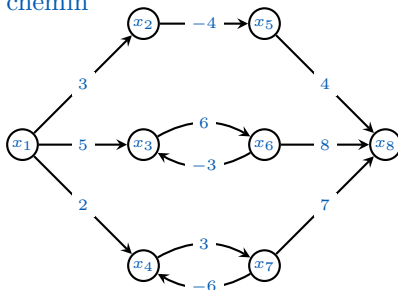
Exemple :



- La longueur du chemin $p_1 = \langle x_1, x_2, x_5, x_8 \rangle$ est $l(p_1)$ avec $l(p_1) = \mathcal{C}_{1,2} + \mathcal{C}_{2,5} + \mathcal{C}_{5,8} = 3 - 4 + 4 = 3$.
- La longueur du chemin $p_2 = \langle x_1, x_3, x_6, x_8 \rangle$ est $l(p_2) = \mathcal{C}_{1,3} + \mathcal{C}_{3,6} + \mathcal{C}_{6,8} = 19$.
- La longueur du chemin $p_3 = \langle x_1, x_3, x_6, x_3, x_6, x_8 \rangle$ est $l(p_3) = 22$.
- La longueur du chemin $p_4 = \langle x_1, x_4, x_7, x_8 \rangle$ est $l(p_4) = 12$.

Poids / longueur d'un chemin

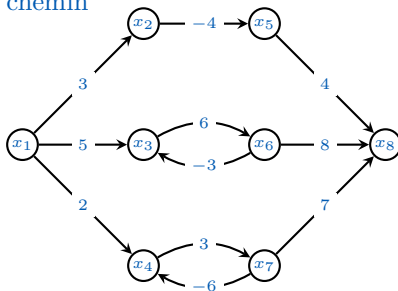
Exemple :



- La longueur du chemin $p_1 = \langle x_1, x_2, x_5, x_8 \rangle$ est $l(p_1)$ avec $l(p_1) = \mathcal{C}_{1,2} + \mathcal{C}_{2,5} + \mathcal{C}_{5,8} = 3 - 4 + 4 = 3$.
- La longueur du chemin $p_2 = \langle x_1, x_3, x_6, x_8 \rangle$ est $l(p_2) = \mathcal{C}_{1,3} + \mathcal{C}_{3,6} + \mathcal{C}_{6,8} = 19$.
- La longueur du chemin $p_3 = \langle x_1, x_3, x_6, x_3, x_6, x_8 \rangle$ est $l(p_3) = 22$.
- La longueur du chemin $p_4 = \langle x_1, x_4, x_7, x_8 \rangle$ est $l(p_4) = 12$.
- La longueur du chemin $p_5 = \langle x_1, x_4, x_7, x_4, x_7, x_8 \rangle$ est $l(p_5) = 9$.

Poids / longueur d'un chemin

Exemple :



- La longueur du chemin $p_1 = \langle x_1, x_2, x_5, x_8 \rangle$ est $l(p_1)$ avec $l(p_1) = \mathcal{C}_{1,2} + \mathcal{C}_{2,5} + \mathcal{C}_{5,8} = 3 - 4 + 4 = 3$.
- La longueur du chemin $p_2 = \langle x_1, x_3, x_6, x_8 \rangle$ est $l(p_2) = \mathcal{C}_{1,3} + \mathcal{C}_{3,6} + \mathcal{C}_{6,8} = 19$.
- La longueur du chemin $p_3 = \langle x_1, x_3, x_6, x_3, x_6, x_8 \rangle$ est $l(p_3) = 22$.
- La longueur du chemin $p_4 = \langle x_1, x_4, x_7, x_8 \rangle$ est $l(p_4) = 12$.
- La longueur du chemin $p_5 = \langle x_1, x_4, x_7, x_4, x_7, x_8 \rangle$ est $l(p_5) = 9$.
- La longueur du chemin $p_6 = \langle x_1, x_4, x_7, x_4, x_7, x_4, x_7, x_8 \rangle$ est $l(p_6) = 6$.



Distance / circuit absorbant

Définition et propriété

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\delta(x_i, x_j)$ **la distance** entre les deux sommets x_i et x_j avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$. Cette distance est définie par :

$$\delta(x_i, x_j) = \begin{cases} \text{la longueur minimal des chemins existants entre } x_i \text{ et } x_j, \\ +\infty \text{ sinon.} \end{cases}$$

Distance / circuit absorbant

Définition et propriété

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\delta(x_i, x_j)$ **la distance** entre les deux sommets x_i et x_j avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$. Cette distance est définie par :

$$\delta(x_i, x_j) = \begin{cases} \text{la longueur minimal des chemins existants entre } x_i \text{ et } x_j, \\ +\infty \text{ sinon.} \end{cases}$$

- S'il existe un chemin entre deux sommets x_i et x_j contenant un circuit de coût total négatif alors $\delta(x_i, x_j) = -\infty$.

Distance / circuit absorbant

Définition et propriété

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\delta(x_i, x_j)$ **la distance** entre les deux sommets x_i et x_j avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$. Cette distance est définie par :

$$\delta(x_i, x_j) = \begin{cases} \text{la longueur minimal des chemins existants entre } x_i \text{ et } x_j, \\ +\infty \text{ sinon.} \end{cases}$$

- S'il existe un chemin entre deux sommets x_i et x_j contenant un circuit de coût total négatif alors $\delta(x_i, x_j) = -\infty$.
- Un circuit de coût négatif est appelé **un circuit absorbant**.

Distance / circuit absorbant

Définition et propriété

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\delta(x_i, x_j)$ **la distance** entre les deux sommets x_i et x_j avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$. Cette distance est définie par :

$$\delta(x_i, x_j) = \begin{cases} \text{la longueur minimal des chemins existants entre } x_i \text{ et } x_j, \\ +\infty \text{ sinon.} \end{cases}$$

- S'il existe un chemin entre deux sommets x_i et x_j contenant un circuit de coût total négatif alors $\delta(x_i, x_j) = -\infty$.
- Un circuit de coût négatif est appelé **un circuit absorbant**.
- Une condition nécessaire d'existence de plus court chemin est l'absence de circuit absorbant.



Distance / circuit absorbant

Définition et propriété

On considère un graphe **orienté valué** $G = (X, A)$ et **une fonction coût** $\mathcal{C} : A \rightarrow \mathbb{R}$.

- On note par $\delta(x_i, x_j)$ **la distance** entre les deux sommets x_i et x_j avec $x_i, x_j \in X$ et $1 \leq i, j \leq |X|$. Cette distance est définie par :

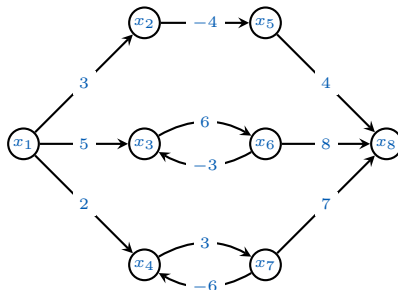
$$\delta(x_i, x_j) = \begin{cases} \text{la longueur minimal des chemins existants entre } x_i \text{ et } x_j, \\ +\infty \text{ sinon.} \end{cases}$$

- S'il existe un chemin entre deux sommets x_i et x_j contenant un circuit de coût total négatif alors $\delta(x_i, x_j) = -\infty$.
- Un circuit de coût négatif est appelé **un circuit absorbant**.
- Une condition nécessaire d'existence de plus court chemin est l'absence de circuit absorbant.
- Le plus court chemin entre deux sommets est composé d'un ensemble de plus courts chemins. La réciproque n'est pas toujours vraie.



Distance / Circuit absorbant

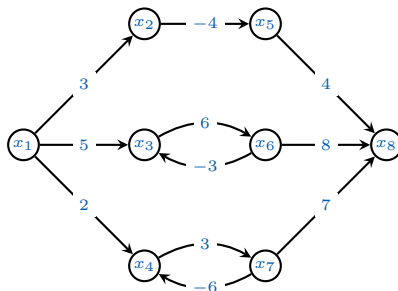
Exemple :





Distance / Circuit absorbant

Exemple :

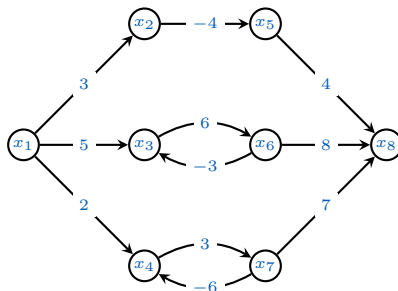


- La distance entre x_1 et x_2 est $\delta(x_1, x_2) = 3$.



Distance / Circuit absorbant

Exemple :

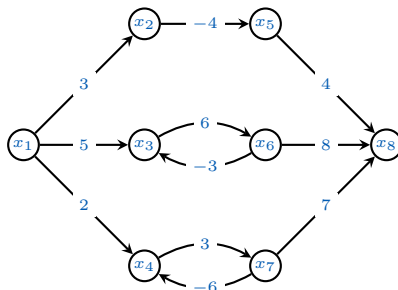


- La distance entre x_1 et x_2 est $\delta(x_1, x_2) = 3$.
- La distance entre x_8 et x_1 est $\delta(x_8, x_1) = +\infty$.



Distance / Circuit absorbant

Exemple :

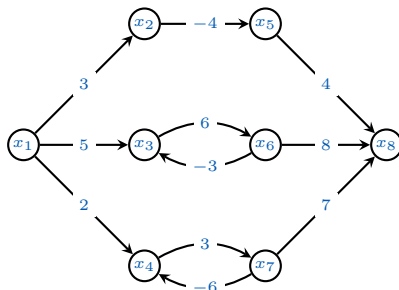


- La distance entre x_1 et x_2 est $\delta(x_1, x_2) = 3$.
- La distance entre x_8 et x_1 est $\delta(x_8, x_1) = +\infty$.
- La distance entre x_1 et x_8 est $\delta(x_1, x_8) = -\infty$: il n'existe pas de plus court chemin entre x_1 et x_8 (existence de circuit absorbant).



Distance / Circuit absorbant

Exemple :



- La distance entre x_1 et x_2 est $\delta(x_1, x_2) = 3$.
- La distance entre x_8 et x_1 est $\delta(x_8, x_1) = +\infty$.
- La distance entre x_1 et x_8 est $\delta(x_1, x_8) = -\infty$: il n'existe pas de plus court chemin entre x_1 et x_8 (existence de circuit absorbant).
- La distance entre x_1 et x_5 est $\delta(x_1, x_5) = \delta(x_1, x_2) + \delta(x_2, x_5) = -1$: deux plus courts chemins.



Recherche du plus court chemin

L'objectif de ce deuxième chapitre est de déterminer le plus court chemin entre deux sommets d'un graphe : déterminer une arborescence optimale. Pour ce faire, nous présentons trois algorithmes utilisés dans des cas différents, à savoir :



Recherche du plus court chemin

L'objectif de ce deuxième chapitre est de déterminer le plus court chemin entre deux sommets d'un graphe : déterminer une arborescence optimale. Pour ce faire, nous présentons trois algorithmes utilisés dans des cas différents, à savoir :

Graphes à coûts positifs : Application de l'algorithme de **DIJKSTRA** :
détermination des plus courts chemins d'une source fixée s à toutes les différentes destinations.

Recherche du plus court chemin

L'objectif de ce deuxième chapitre est de déterminer le plus court chemin entre deux sommets d'un graphe : déterminer une arborescence optimale. Pour ce faire, nous présentons trois algorithmes utilisés dans des cas différents, à savoir :

Graphes à coûts positifs : Application de l'algorithme de **DIJKSTRA** :
détermination des plus courts chemins d'une source fixée s à toutes
les différentes destinations.

Graphes à coûts quelconques : Application de l'algorithme de **FORD-BELLMAN** :
détermination des plus courts chemins d'une source fixée s à toutes
les différentes destinations.



Recherche du plus court chemin

L'objectif de ce deuxième chapitre est de déterminer le plus court chemin entre deux sommets d'un graphe : déterminer une arborescence optimale. Pour ce faire, nous présentons trois algorithmes utilisés dans des cas différents, à savoir :

Graphes à coûts positifs : Application de l'algorithme de **DIJKSTRA** :
détermination des plus courts chemins d'une source fixée s à toutes les différentes destinations.

Graphes à coûts quelconques : Application de l'algorithme de **FORD-BELLMAN** :
détermination des plus courts chemins d'une source fixée s à toutes les différentes destinations.

Graphes à coûts quelconques : Application de l'algorithme de **FLOYD-WARSHALL** : une généralisation de l'algorithme de FORD-BELLMAN. Détermination d'un plus court chemin entre n'importe quel couple de sommets du graphe.

1 Introduction

2 Algorithme de DIJKSTRA

- Principe
- Application
- Limitation

3 Algorithme de FORD BELLMAN

4 Algorithme de FLOYD-WARSHALL



Pseudo-code de l'algorithme de DIJKSTRA

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$ et $\mathcal{C} : A \rightarrow \mathbb{R}$, la fonction coût associée.

Notation :

- Pour tout $j \in \{1, \dots, n\}$, on note par $dist(j)$: la longueur optimale du chemin reliant la source x_1 à la destination x_j . Cette longueur sera mise à jour tout au long du processus afin de déterminer $\delta(x_1, x_j)$, le plus court chemin reliant les deux sommets.
- Pour tout $j \in \{1, \dots, n\}$, on note par $Pcc(j)$: le plus court chemin associé au sommet x_j .
- On note par P la liste des sommets dont les $dist$ sont fixées de manière Permanente : pour lesquels le plus court chemin a été déterminé.
- On note par T la liste des sommets dont les $dist$ sont fixées de manière Temporaire : pour lesquels le plus court chemin n'a pas encore été déterminé.



Pseudo-code de l'algorithme de DIJKSTRA

Pseudo-code

Initialisation :

- $dist(1) = 0$
- Pour j de 2 à n faire

$$dist(j) = \begin{cases} C_{1,j} & \text{si } (x_1, x_j) \in A, \\ +\infty & \text{sinon.} \end{cases}$$
- fin pour
- $Pcc(1) = (x_1)$, $P = \{x_1\}$, et $T = \{x_2, \dots, x_n\}$

Procédure itérative :

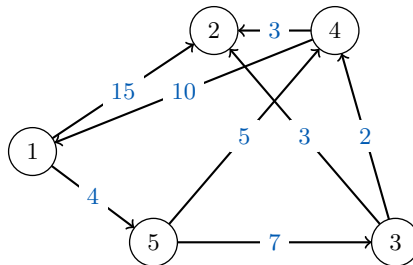
- Tant que $T \neq \emptyset$ faire
 - étape 1 : Choix de $dist(k)$ optimale
 - Détermination de x_k / $dist(k) = \min_{x_j \in T} dist(j)$
 - Mémorisation de $Pcc(k)$, $P = P \cup \{x_k\}$ et $T = T / \{x_k\}$
 - étape 2 : Mise à jour des $dist$
 - Pour $x_j \in T$ et x_j successeur de x_k faire
 - Si $dist(k) + C_{k,j} < dist(j)$ alors

$$dist(j) = dist(k) + C_{k,j}$$
 - fin Si
 - fin Pour
- fin Tant que
- Afficher $dist$ et Pcc .





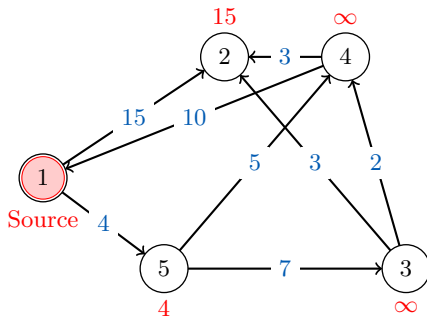
Illustration de l'algorithme de DIJKSTRA



		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2										
3	3										
4	4										
5	5										
P											
T											

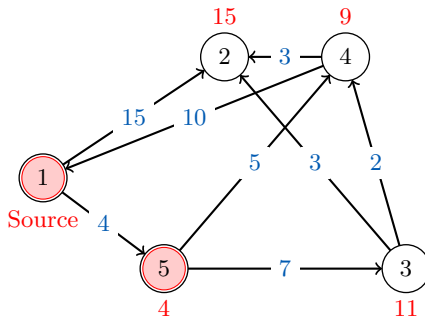
Principe

Illustration de l'algorithme de DIJKSTRA



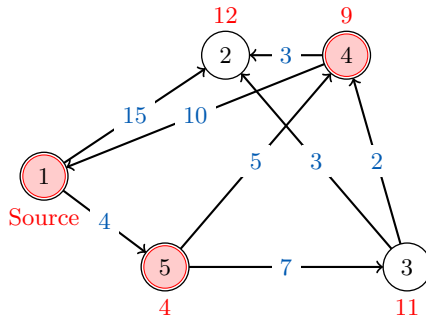
	Itération 0			Itération 1		Itération 2		Itération 3		Itération 4	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	15	(1, 2)								
3	3	∞	(1, 3)								
4	4	∞	(1, 4)								
5	5	4	(1, 5)								
P	{1}										
T	{2, 3, 4, 5}										

Illustration de l'algorithme de DIJKSTRA



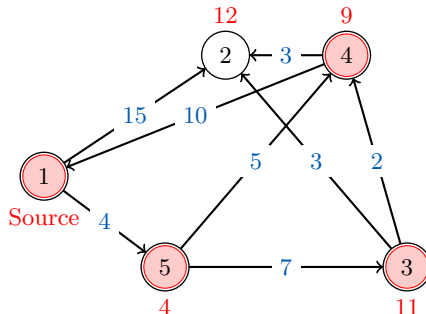
		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	15	(1, 2)	15	(1, 2)						
3	3	∞	(1, 3)	11	(1, 5, 3)						
4	4	∞	(1, 4)	9	(1, 5, 4)						
5	5	4	(1, 5)	—	—	—	—	—	—	—	—
P		{1}		{1, 5}							
T		{2, 3, 4, 5}		{2, 3, 4}							

Illustration de l'algorithme de DIJKSTRA



		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	15	(1, 2)	15	(1, 2)	12	(1, 5, 4, 2)				
3	3	∞	(1, 3)	11	(1, 5, 3)	11	(1, 5, 3)				
4	4	∞	(1, 4)	9	(1, 5, 4)	—	—	—	—	—	—
5	5	4	(1, 5)	—	—	—	—	—	—	—	—
P	{1}			{1, 5}			{1, 5, 4}			HONORIS UNITED UNIVERSITIES	
T	{2, 3, 4, 5}			{2, 3, 4}			{2, 3}				

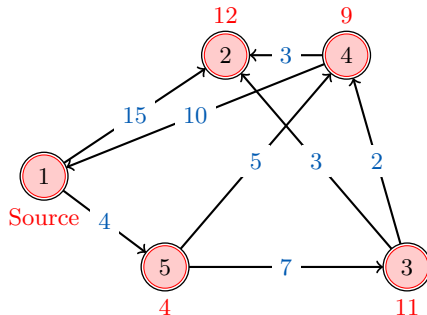
Illustration de l'algorithme de DIJKSTRA



		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	15	(1, 2)	15	(1, 2)	12	(1, 5, 4, 2)				
3	3	∞	(1, 3)	11	(1, 5, 3)	11	(1, 5, 3)	—	—	—	—
4	4	∞	(1, 4)	9	(1, 5, 4)	—	—	—	—	—	—
5	5	4	(1, 5)	—	—	—	—	—	—	—	—
P	{1}			{1, 5}			{1, 5, 4}			{1, 5, 4, 3}	
T	{2, 3, 4, 5}			{2, 3, 4}			{2, 3}			{2}	



Illustration de l'algorithme de DIJKSTRA

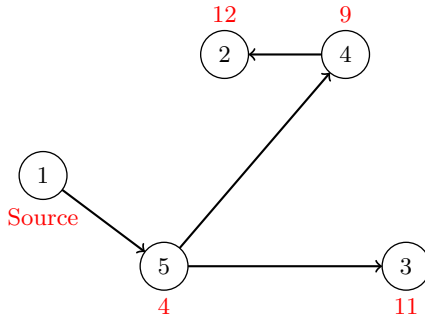


		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	15	(1, 2)	15	(1, 2)	12	(1, 5, 4, 2)	12	(1, 5, 4, 2)	—	—
3	3	∞	(1, 3)	11	(1, 5, 3)	11	(1, 5, 3)	—	—	—	—
4	4	∞	(1, 4)	9	(1, 5, 4)	—	—	—	—	—	—
5	5	4	(1, 5)	—	—	—	—	—	—	—	—
P	{1}			{1, 5}			{1, 5, 4}			{1, 5, 4, 3}	
T	{2, 3, 4, 5}			{2, 3, 4}			{2, 3}			{2}	



Illustration de l'algorithme de DIJKSTRA

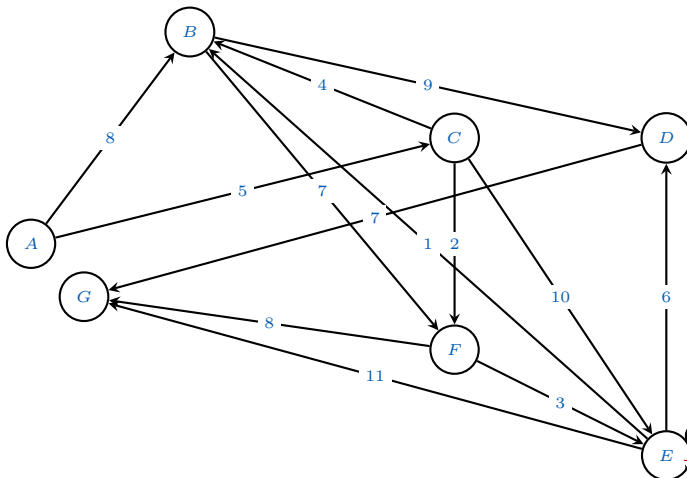
Arborescence optimale



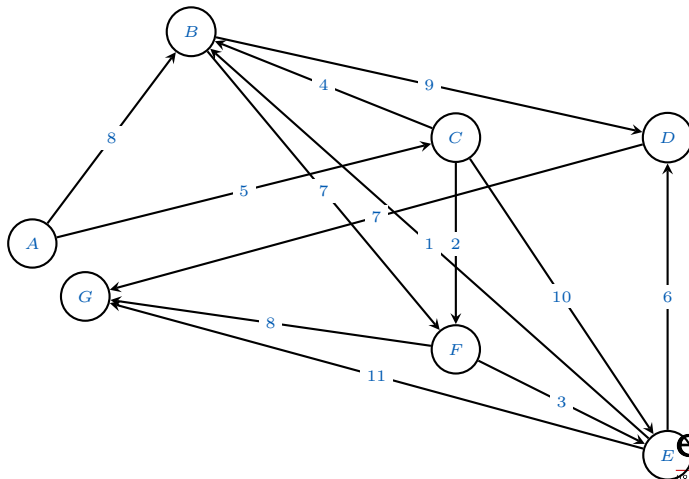
15 / 78

Application de l'algorithme de DIJKSTRA

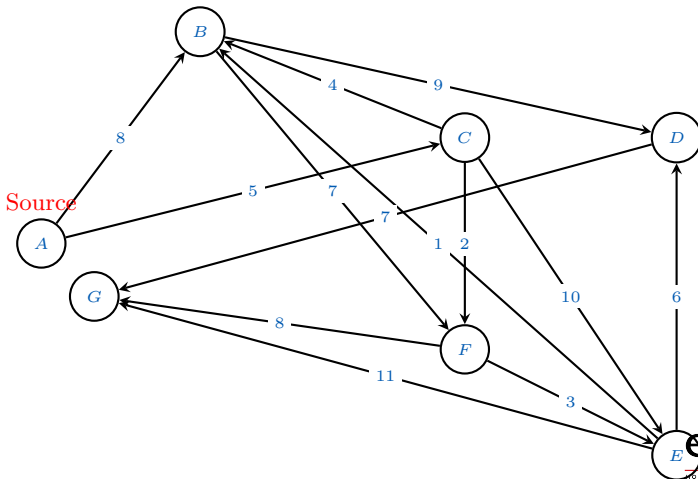
Représentation graphique du problème :



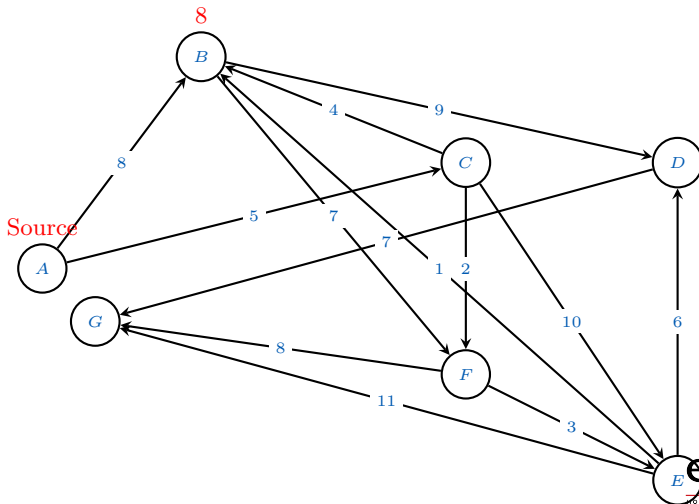
Application de l'algorithme de DIJKSTRA : Itération 0



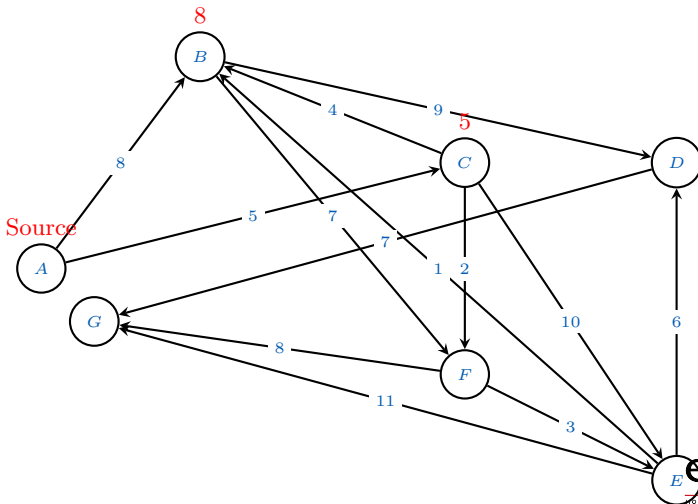
Application de l'algorithme de DIJKSTRA : Itération 0



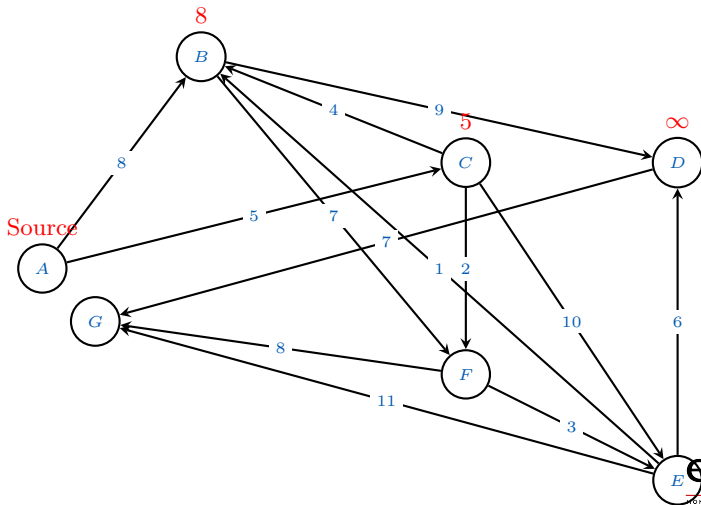
Application de l'algorithme de DIJKSTRA : Itération 0



Application de l'algorithme de DIJKSTRA : Itération 0

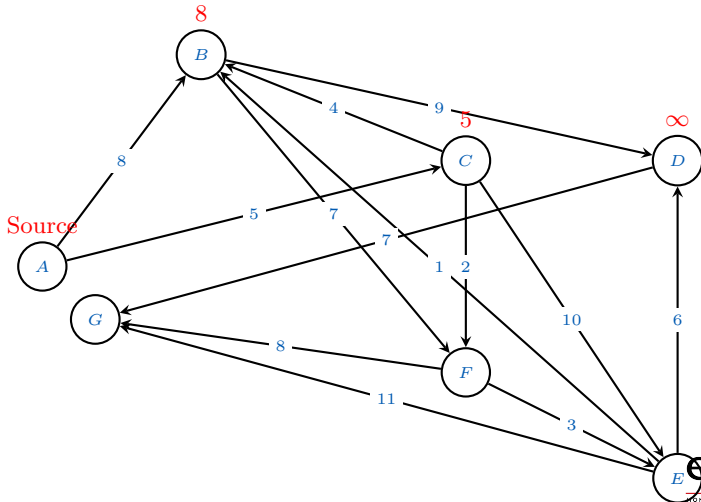


Application de l'algorithme de DIJKSTRA : Itération 0

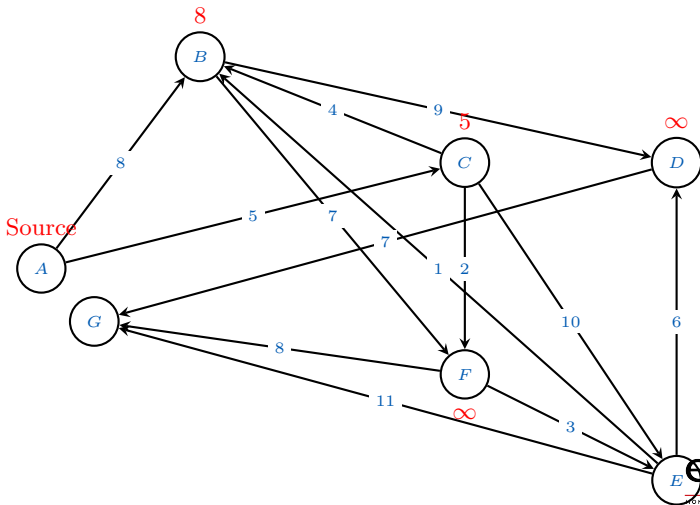


Application

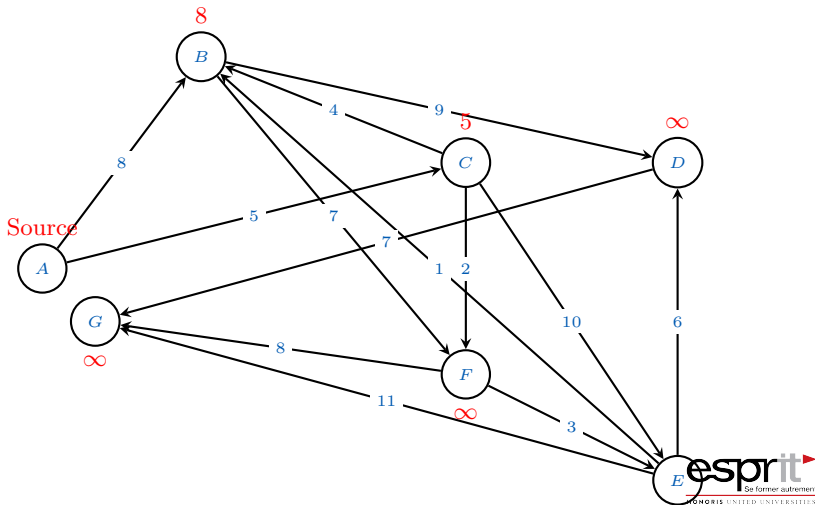
Application de l'algorithme de DIJKSTRA : Itération 0



Application de l'algorithme de DIJKSTRA : Itération 0



Application de l'algorithme de DIJKSTRA : Itération 0

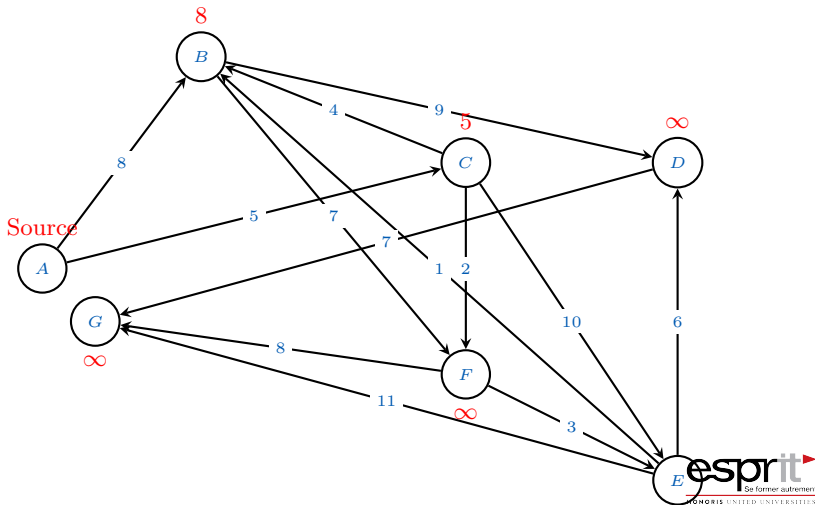




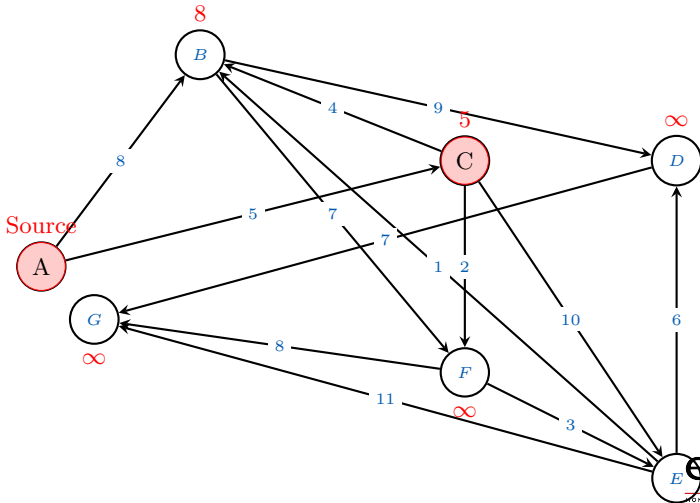
Application de l'algorithme de DIJKSTRA : Itération 0

j	x_j	Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5		Itération 6	
		$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	B	8	(A,B)												
3	C	5	(A,C)												
4	D	∞	(A,D)												
5	E	∞	(A,E)												
6	F	∞	(A,F)												
7	G	∞	(A,G)												
P	{A}														
T	{B,C,D,E,F,G}														

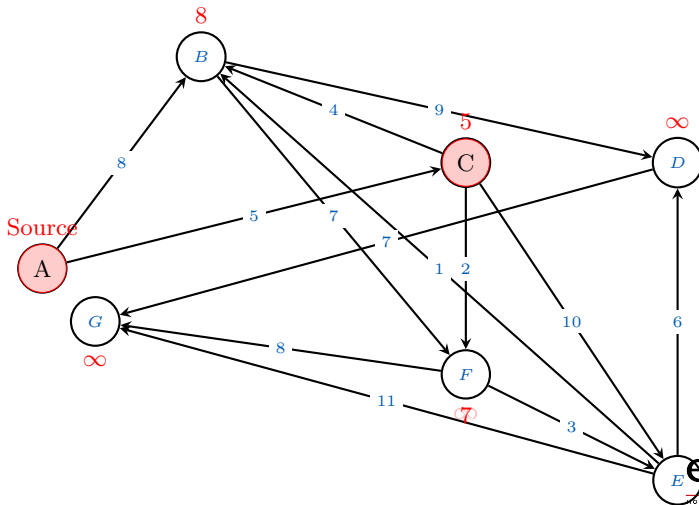
Application de l'algorithme de DIJKSTRA : Itération 1



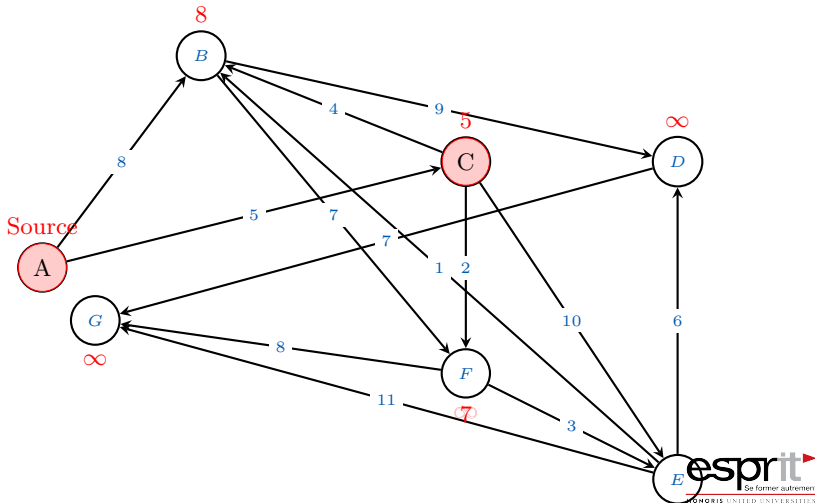
Application de l'algorithme de DIJKSTRA : Itération 1



Application de l'algorithme de DIJKSTRA : Itération 1



Application de l'algorithme de DIJKSTRA : Itération 1

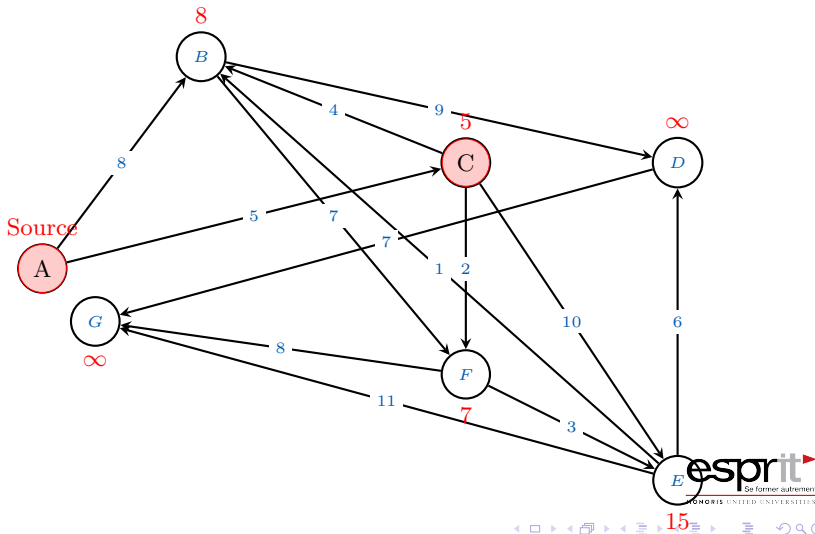




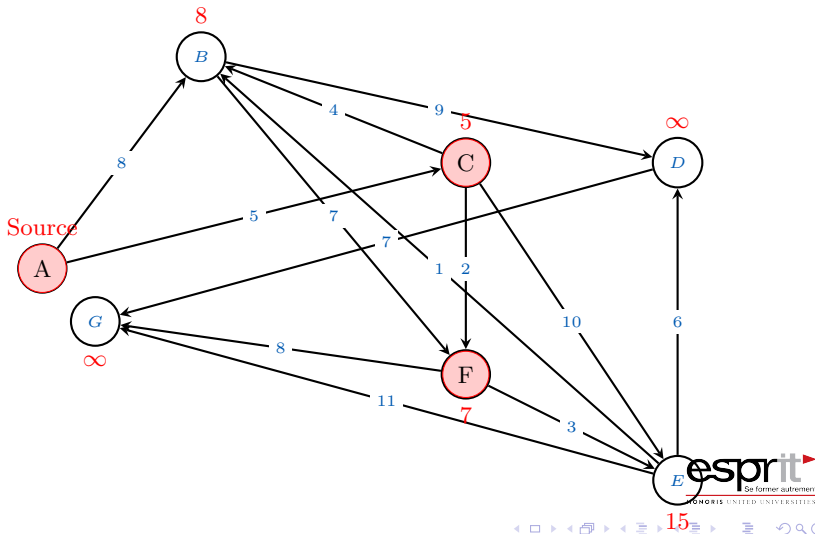
Application de l'algorithme de DIJKSTRA : Itération 1

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5		Itération 6	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	B	8	(A,B)	8	(A,B)										
3	C	5	(A,C)	-	-	-	-	-	-	-	-	-	-	-	
4	D	∞	(A,D)	∞	(A,D)										
5	E	∞	(A,E)	15	(A,C,E)										
6	F	∞	(A,F)	7	(A,C,F)										
7	G	∞	(A,G)	∞	(A,G)										
P		{A}		{A,C}											
T		{B,C,D,E,F,G}		{B,D,E,F,G}											

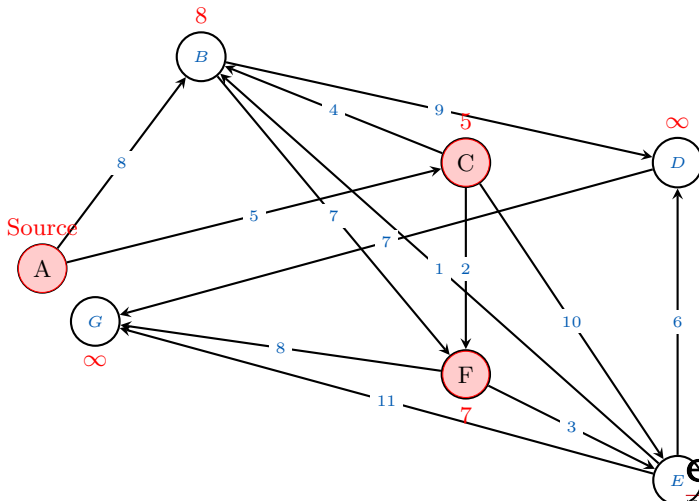
Application de l'algorithme de DIJKSTRA : Itération 2



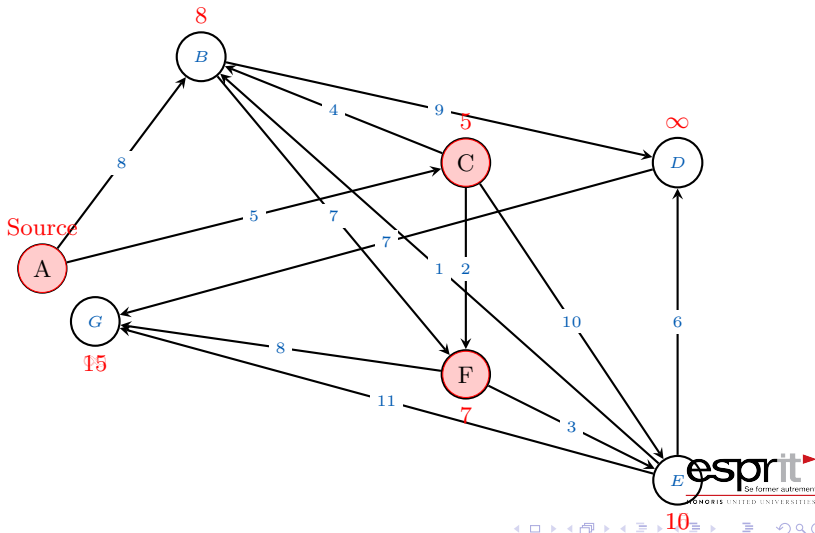
Application de l'algorithme de DIJKSTRA : Itération 2



Application de l'algorithme de DIJKSTRA : Itération 2



Application de l'algorithme de DIJKSTRA : Itération 2



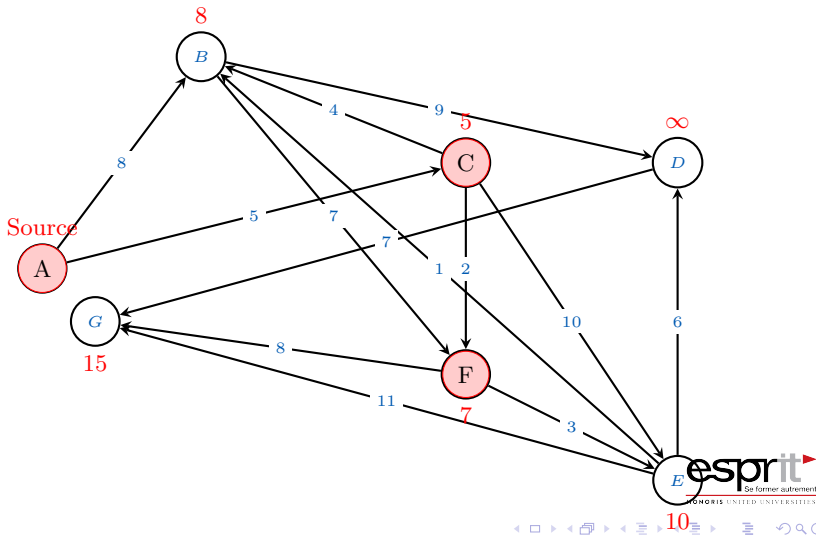


Application

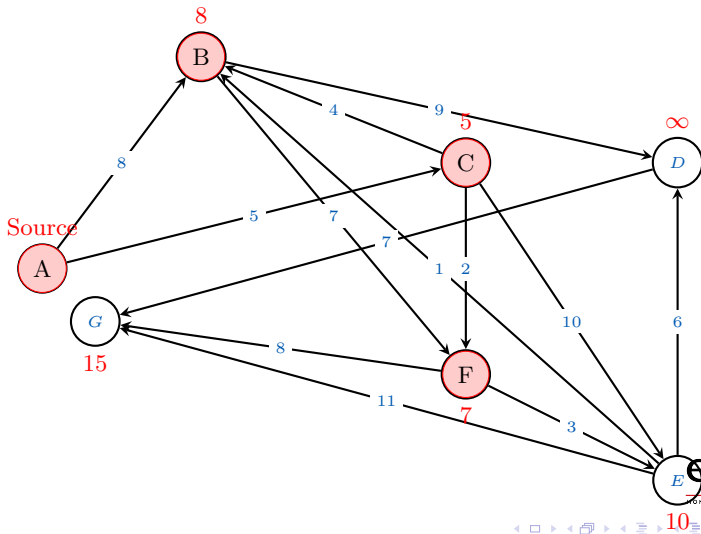
Application de l'algorithme de DIJKSTRA : Itération 2

j	x_j	Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5		Itération 6	
		$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	B	8	(A,B)	8	(A,B)	8	(A,B)								
3	C	5	(A,C)	-	-	-	-	-	-	-	-	-	-	-	
4	D	∞	(A,D)	∞	(A,D)	∞	(A,D)								
5	E	∞	(A,E)	15	(A,C,E)	10	(A,C,F,E)								
6	F	∞	(A,F)	7	(A,C,F)	-	-	-	-	-	-	-	-	-	
7	G	∞	(A,G)	∞	(A,G)	15	(A,C,F,G)								
P	{A}			{A,C}			{A,C,F}								
T	{B,C,D,E,F,G}			{B,D,E,F,G}			{B,D,E,G}								

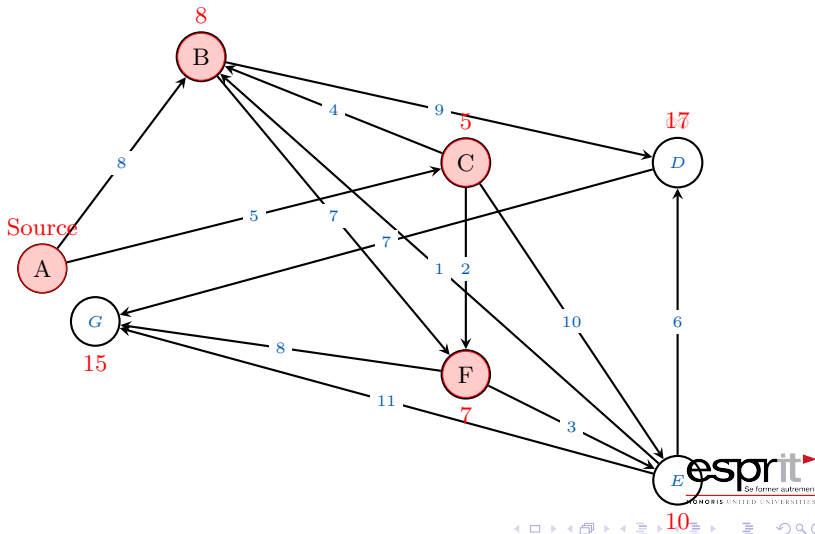
Application de l'algorithme de DIJKSTRA : Itération 3



Application de l'algorithme de DIJKSTRA : Itération 3



Application de l'algorithme de DIJKSTRA : Itération 3



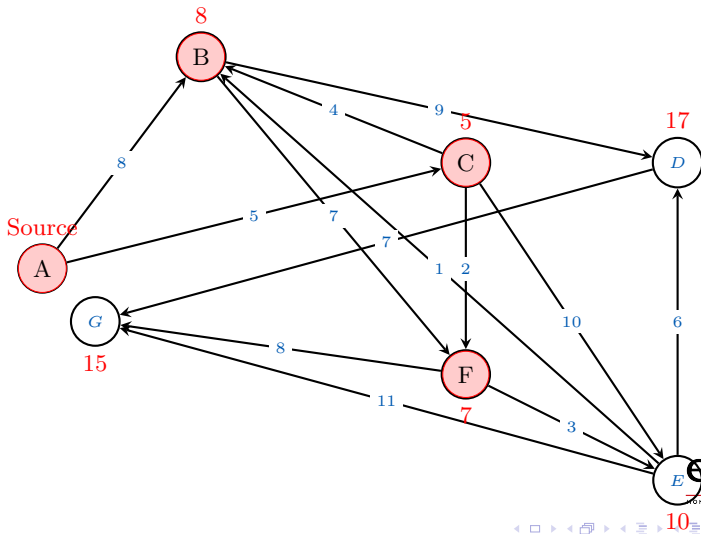


Application

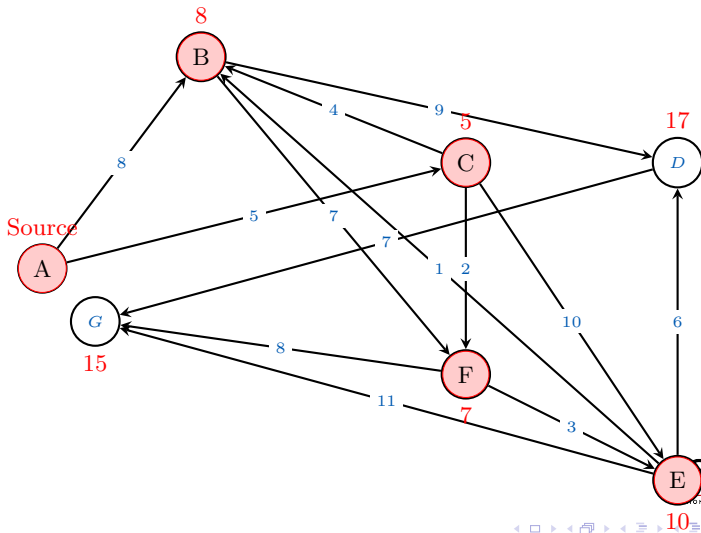
Application de l'algorithme de DIJKSTRA : Itération 3

j	x_j	Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5		Itération 6	
		$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	B	8	(A,B)	8	(A,B)	8	(A,B)	-	-	-	-	-	-	-	-
3	C	5	(A,C)	-	-	-	-	-	-	-	-	-	-	-	-
4	D	∞	(A,D)	∞	(A,D)	∞	(A,D)	17	(A,B,D)						
5	E	∞	(A,E)	15	(A,C,E)	10	(A,C,F,E)	10	(A,C,F,E)						
6	F	∞	(A,F)	7	(A,C,F)	-	-	-	-	-	-	-	-	-	-
7	G	∞	(A,G)	∞	(A,G)	15	(A,C,F,G)	15	(A,C,F,G)						
P		{A}		{A,C}		{A,C,F}		{A,C,F,B}							
T		{B,C,D,E,F,G}		{B,D,E,F,G}		{B,D,E,G}		{D,E,G}							

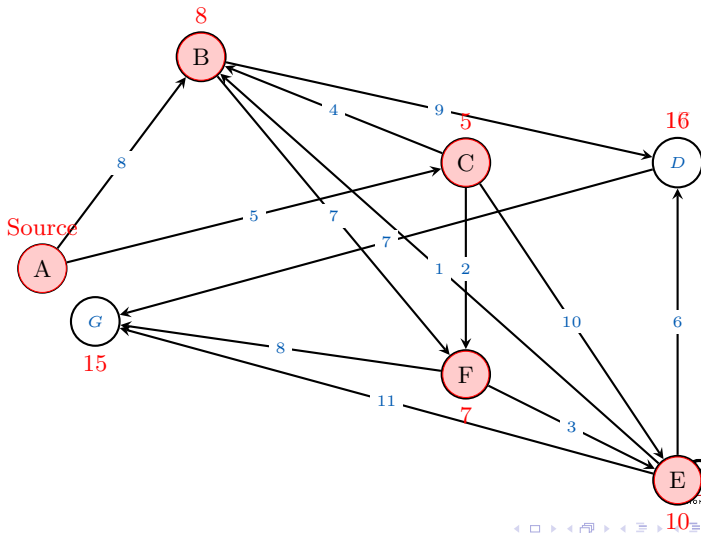
Application de l'algorithme de DIJKSTRA : Itération 4



Application de l'algorithme de DIJKSTRA : Itération 4



Application de l'algorithme de DIJKSTRA : Itération 4



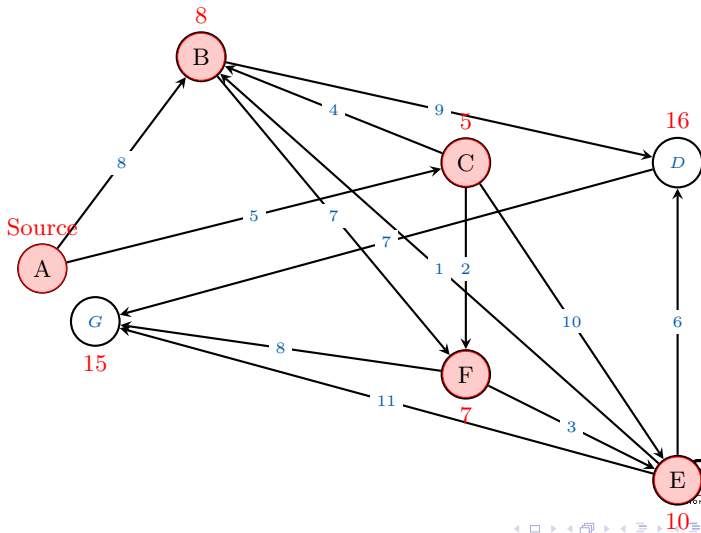


Application de l'algorithme de DIJKSTRA : Itération 4

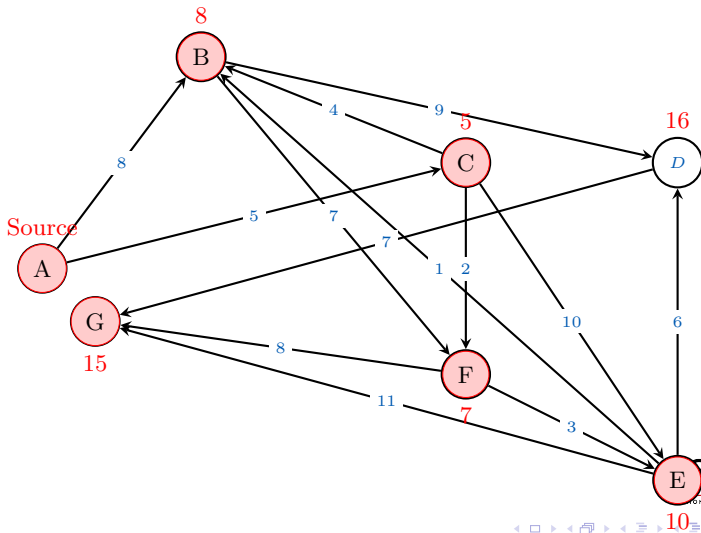


		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5		Itération 6	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	B	8	(A,B)	8	(A,B)	8	(A,B)	-	-	-	-	-	-	-	-
3	C	5	(A,C)	-	-	-	-	-	-	-	-	-	-	-	-
4	D	∞	(A,D)	∞	(A,D)	∞	(A,D)	17	(A,B,D)	16	(A,C,F,E,D)				
5	E	∞	(A,E)	15	(A,C,E)	10	(A,C,F,E)	10	(A,C,F,E)	-	-	-	-	-	-
6	F	∞	(A,F)	7	(A,C,F)	-	-	-	-	-	-	-	-	-	-
7	G	∞	(A,G)	∞	(A,G)	15	(A,C,F,G)	15	(A,C,F,G)	15	(A,C,F,G)				
P		{A}		{A,C}		{A,C,F}		{A,C,F,B}		{A,C,F,B,E}					
T		{B,C,D,E,F,G}		{B,D,E,F,G}		{B,D,E,G}		{D,E,G}		{D,G}					

Application de l'algorithme de DIJKSTRA : Itération 5



Application de l'algorithme de DIJKSTRA : Itération 5





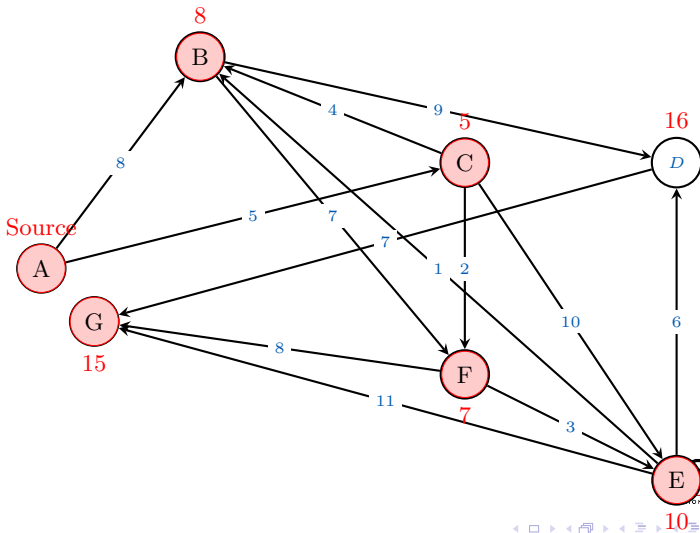
Application de l'algorithme de DIJKSTRA : Itération 5



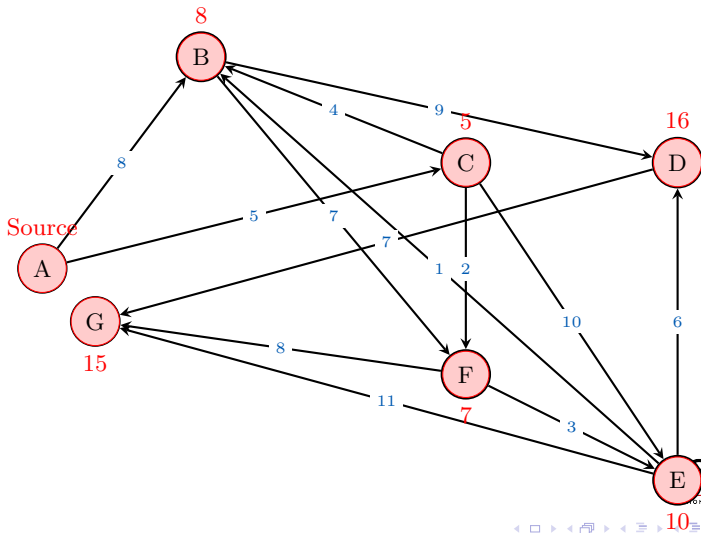
		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5		Itération 6	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	B	8	(A,B)	8	(A,B)	8	(A,B)	-	-	-	-	-	-	-	-
3	C	5	(A,C)	-	-	-	-	-	-	-	-	-	-	-	-
4	D	∞	(A,D)	∞	(A,D)	∞	(A,D)	17	(A,B,D)	16	(A,C,F,E,D)	16	(A,C,F,E,D)	-	-
5	E	∞	(A,E)	15	(A,C,E)	10	(A,C,F,E)	10	(A,C,F,E)	-	-	-	-	-	-
6	F	∞	(A,F)	7	(A,C,F)	-	-	-	-	-	-	-	-	-	-
7	G	∞	(A,G)	∞	(A,G)	15	(A,C,F,G)	15	(A,C,F,G)	15	(A,C,F,G)	-	-	-	-
P		{A}		{A,C}		{A,C,F}		{A,C,F,B}		{A,C,F,B,E}		{A,C,F,B,E,G}			
T		{B,C,D,E,F,G}		{B,D,E,F,G}		{B,D,E,G}		{D,E,G}		{D,G}		{D}			

Application

Application de l'algorithme de DIJKSTRA : Itération 6



Application de l'algorithme de DIJKSTRA : Itération 6





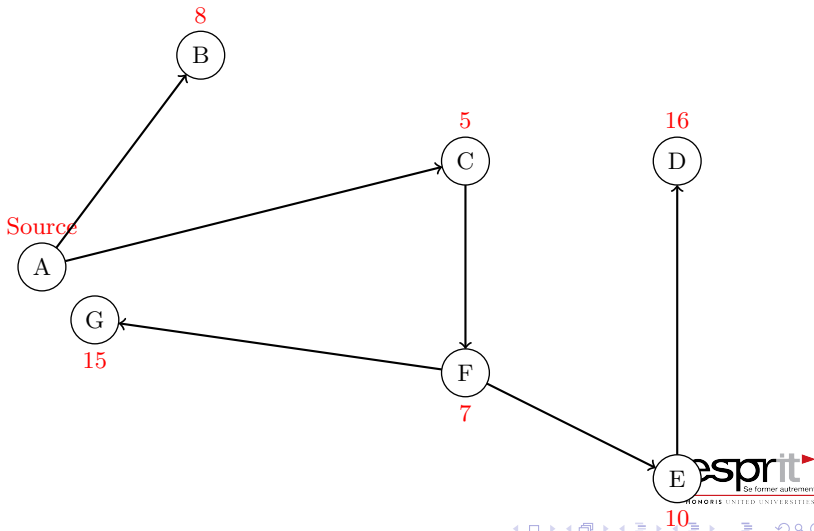
Application de l'algorithme de DIJKSTRA : Itération 6



		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5		Itération 6	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	B	8	(A,B)	8	(A,B)	8	(A,B)	-	-	-	-	-	-	-	-
3	C	5	(A,C)	-	-	-	-	-	-	-	-	-	-	-	-
4	D	∞	(A,D)	∞	(A,D)	∞	(A,D)	17	(A,B,D)	16	(A,C,F,E,D)	16	(A,C,F,E,D)	-	-
5	E	∞	(A,E)	15	(A,C,E)	10	(A,C,F,E)	10	(A,C,F,E)	-	-	-	-	-	-
6	F	∞	(A,F)	7	(A,C,F)	-	-	-	-	-	-	-	-	-	-
7	G	∞	(A,G)	∞	(A,G)	15	(A,C,F,G)	15	(A,C,F,G)	15	(A,C,F,G)	-	-	-	-
P		{A}		{A,C}		{A,C,F}		{A,C,F,B}		{A,C,F,B,E}		{A,C,F,B,E,G}		{A,C,F,B,E,G,D}	
T		{B,C,D,E,F,G}		{B,D,E,F,G}		{B,D,E,G}		{D,E,G}		{D,G}		{D}		\emptyset	

Application

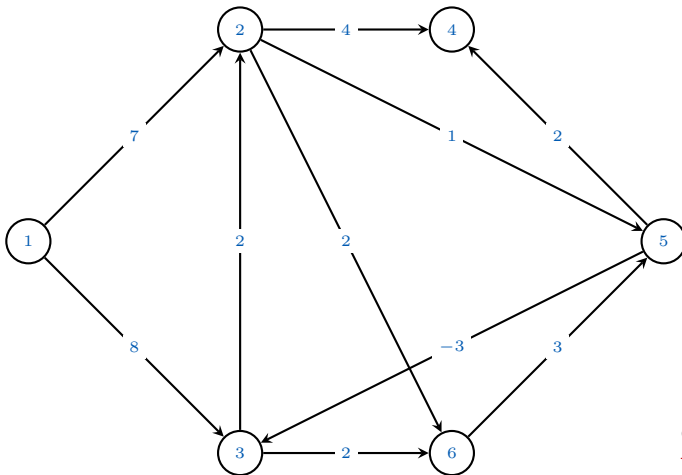
Application de l'algorithme de DIJKSTRA : Arborescence optimale



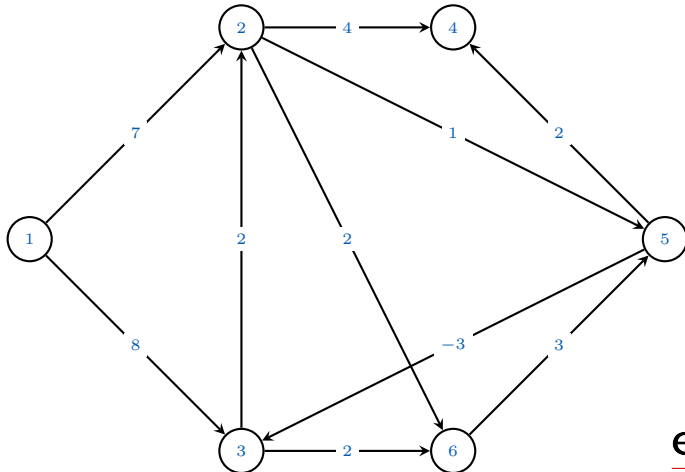
Limitation

Limitation de l'algorithme de DIJKSTRA

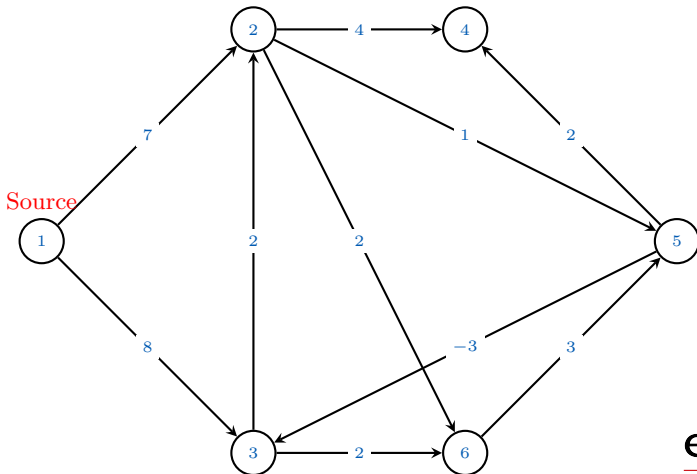
Application de l'algorithme de DIJKSTRA sur un graphe valué à poids mixtes



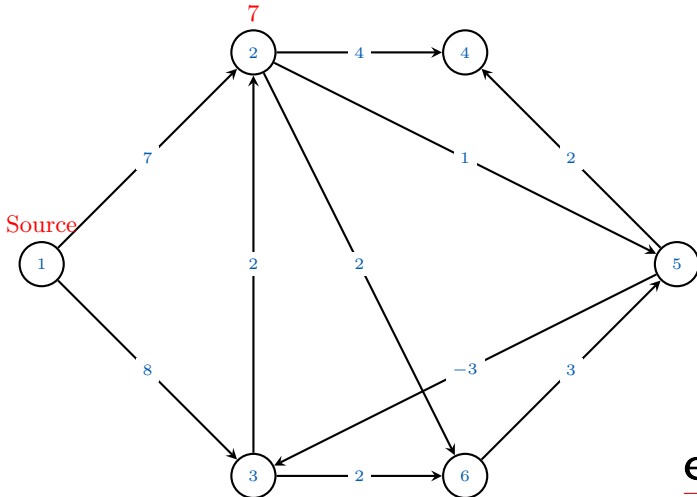
Application de l'algorithme de DIJKSTRA : Itération 0



Application de l'algorithme de DIJKSTRA : Itération 0

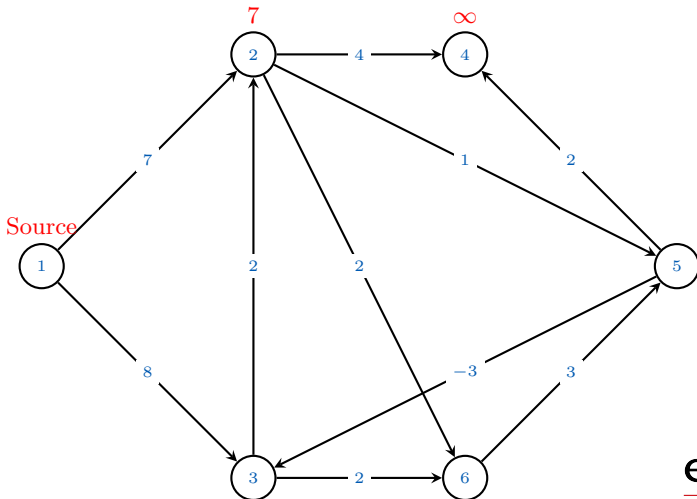


Application de l'algorithme de DIJKSTRA : Itération 0



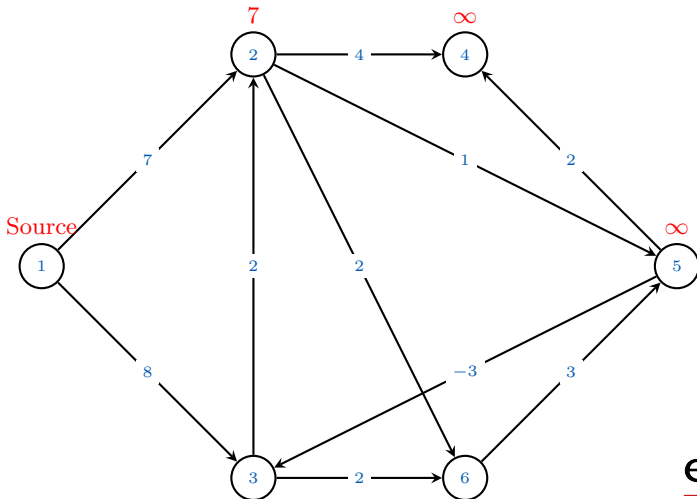


Application de l'algorithme de DIJKSTRA : Itération 0

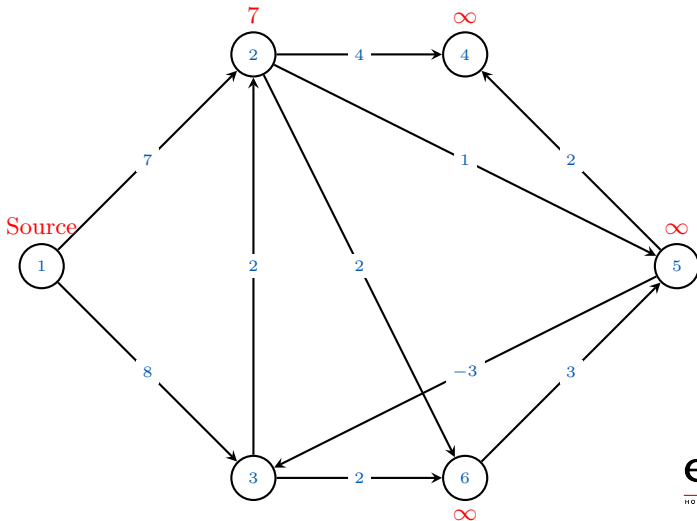




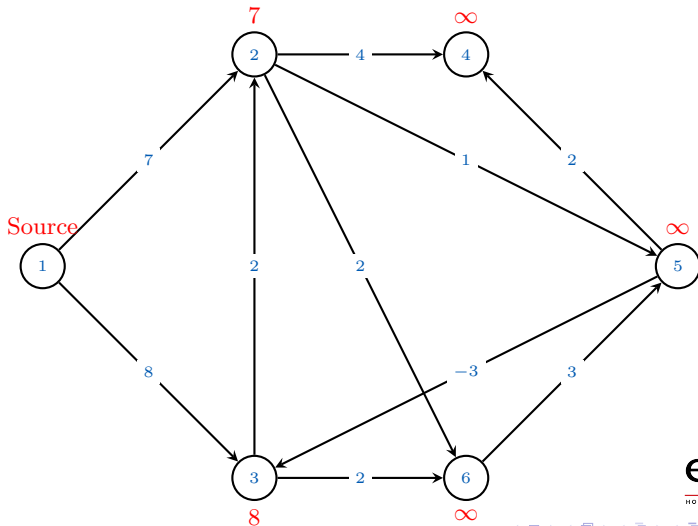
Application de l'algorithme de DIJKSTRA : Itération 0



Application de l'algorithme de DIJKSTRA : Itération 0



Application de l'algorithme de DIJKSTRA : Itération 0



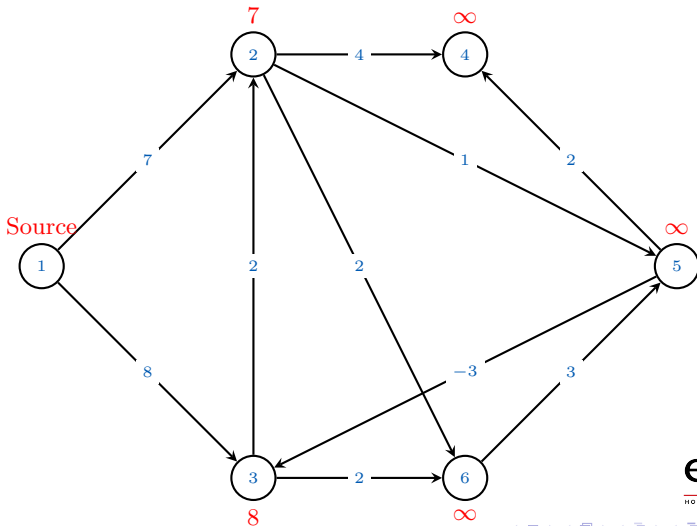


Limitation

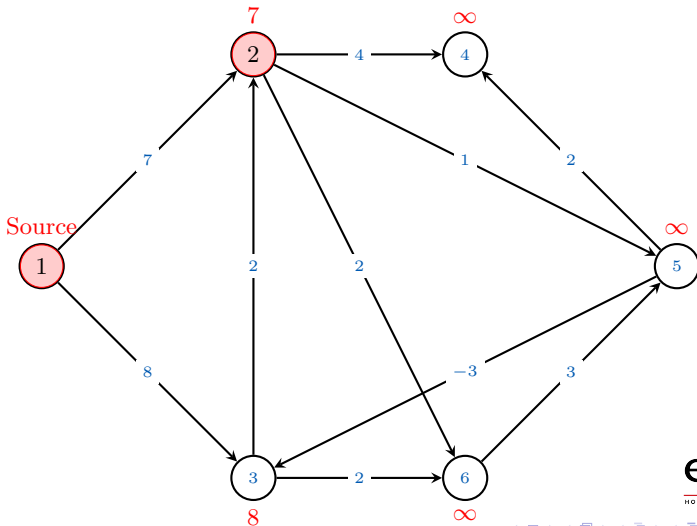
Application de l'algorithme de DIJKSTRA : Itération 0

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	7	(1,2)										
3	3	8	(1,3)										
4	4	∞	(1,4)										
5	5	∞	(1,5)										
6	6	∞	(1,6)										
P	{1}												
T	{2,3,4,5,6}												

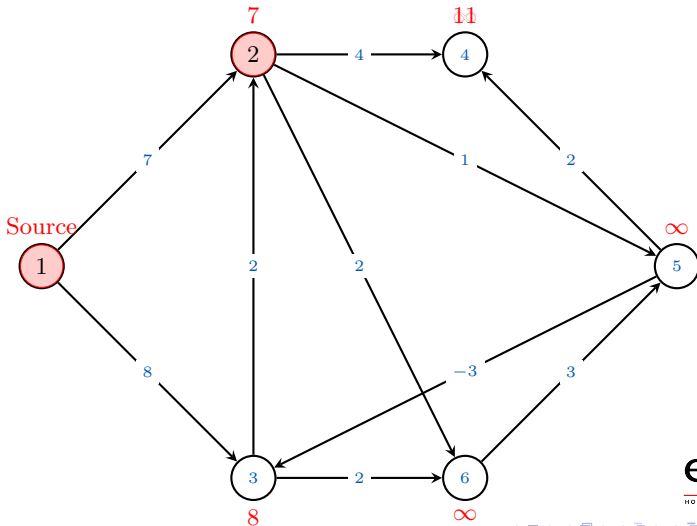
Application de l'algorithme de DIJKSTRA : Itération 1



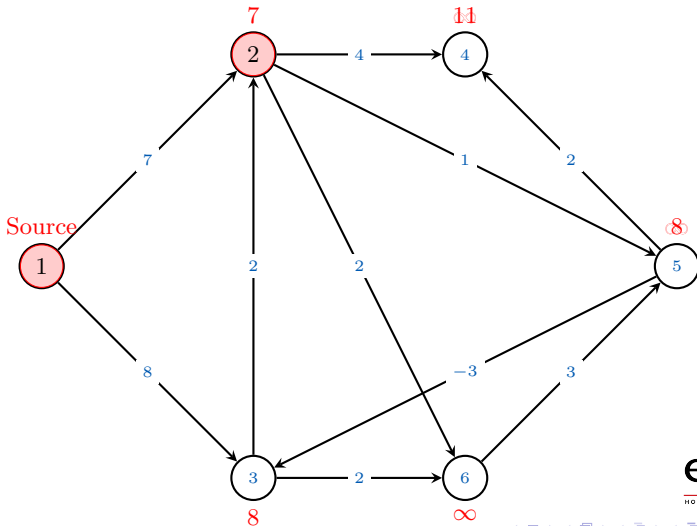
Application de l'algorithme de DIJKSTRA : Itération 1



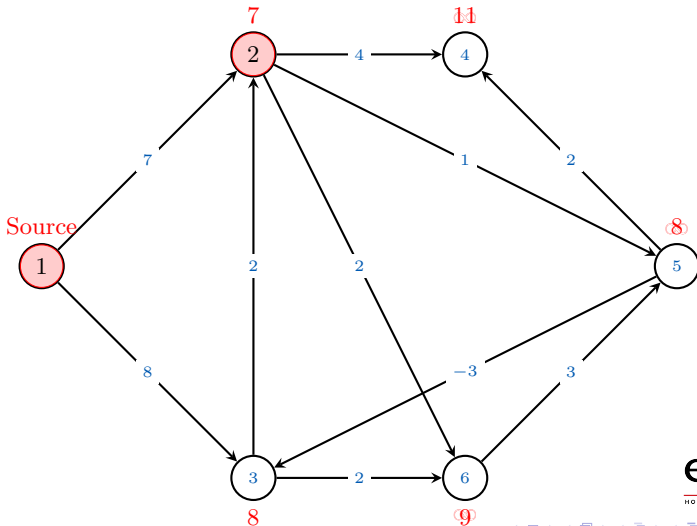
Application de l'algorithme de DIJKSTRA : Itération 1



Application de l'algorithme de DIJKSTRA : Itération 1



Application de l'algorithme de DIJKSTRA : Itération 1



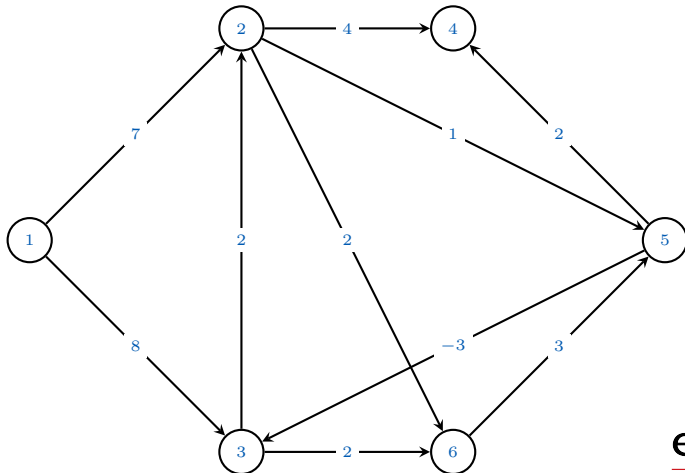


Limitation

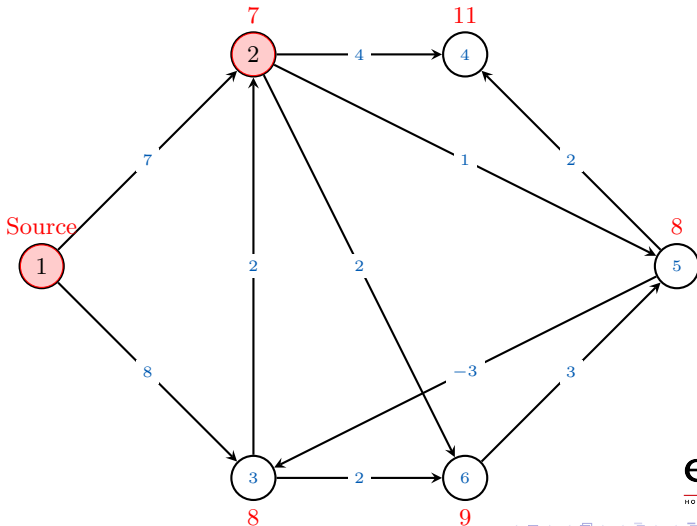
Application de l'algorithme de DIJKSTRA : Itération 1

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	7	(1,2)	-	-	-	-	-	-	-	-	-	
3	3	8	(1,3)	8	(1,3)								
4	4	∞	(1,4)	11	(1,2,4)								
5	5	∞	(1,5)	8	(1,2,5)								
6	6	∞	(1,6)	9	(1,2,6)								
P	{1}			{1,2}									
T	{2,3,4,5,6}			{3,4,5,6}									

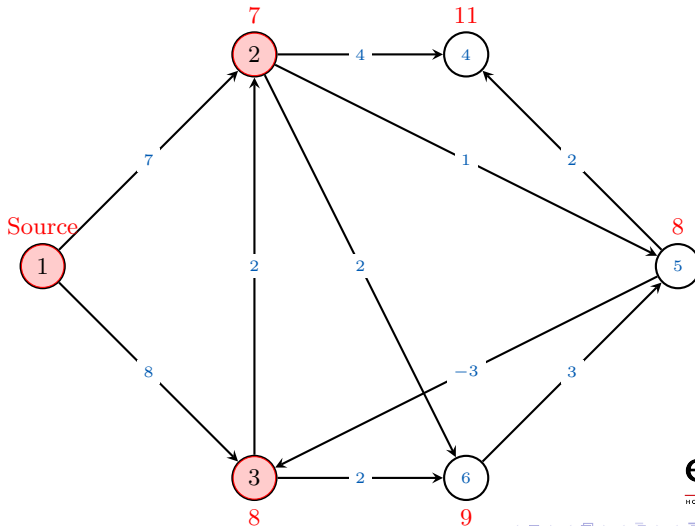
Application de l'algorithme de DIJKSTRA : Itération 2



Application de l'algorithme de DIJKSTRA : Itération 2



Application de l'algorithme de DIJKSTRA : Itération 2





Application de l'algorithme de DIJKSTRA : Itération 2

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	7	(1,2)	-	-	-	-	-	-	-	-	-	-
3	3	8	(1,3)	8	(1,3)	-	-	-	-	-	-	-	-
4	4	∞	(1,4)	11	(1,2,4)	11	(1,2,4)						
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)						
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)						
P	{1}			{1,2}		{1,2,3}							
T	{2,3,4,5,6}			{3,4,5,6}		{4,5,6}							

Application de l'algorithme de DIJKSTRA : Itération 2

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	7	(1,2)	-	-	-	-	-	-	-	-	-	-
3	3	8	(1,3)	8	(1,3)	-	-	-	-	-	-	-	-
4	4	∞	(1,4)	11	(1,2,4)	11	(1,2,4)						
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)						
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)						
P	{1}			{1,2}		{1,2,3}							
T	{2,3,4,5,6}			{3,4,5,6}		{4,5,6}							

- L'algorithme de DIJKSTRA n'a pas détecté le plus court chemin $p = < 1, 2, 5, 3 >$ pour aller du sommet 1 vers le sommet 3.



Application de l'algorithme de DIJKSTRA : Itération 2

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	7	(1,2)	-	-	-	-	-	-	-	-	-	-
3	3	8	(1,3)	8	(1,3)	-	-	-	-	-	-	-	-
4	4	∞	(1,4)	11	(1,2,4)	11	(1,2,4)						
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)						
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)						
P	{1}			{1,2}		{1,2,3}							
T	{2,3,4,5,6}			{3,4,5,6}		{4,5,6}							

- L'algorithme de DIJKSTRA n'a pas détecté le plus court chemin $p = < 1, 2, 5, 3 >$ pour aller du sommet 1 vers le sommet 3.
- $l(p) = 5$ alors que L'algorithme de DIJKSTRA nous a retourné $\delta(1, 3) = 8$.

Application de l'algorithme de DIJKSTRA : Itération 2

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	7	(1,2)	-	-	-	-	-	-	-	-	-	-
3	3	8	(1,3)	8	(1,3)	-	-	-	-	-	-	-	-
4	4	∞	(1,4)	11	(1,2,4)	11	(1,2,4)						
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)						
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)						
P	{1}			{1,2}			{1,2,3}						
T	{2,3,4,5,6}			{3,4,5,6}			{4,5,6}						

- L'algorithme de DIJKSTRA n'a pas détecté le plus court chemin $p = \langle 1, 2, 5, 3 \rangle$ pour aller du sommet 1 vers le sommet 3.
- $l(p) = 5$ alors que L'algorithme de DIJKSTRA nous a retourné $\delta(1, 3) = 8$.
- L'algorithme DIJKSTRA n'est plus utilisable dans le cas d'un graphe valué à poids mixtes.

Application de l'algorithme de DIJKSTRA : Itération 2

		Itération 0		Itération 1		Itération 2		Itération 3		Itération 4		Itération 5	
j	x_j	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$	$dist(x_j)$	$Pcc(x_j)$
2	2	7	(1,2)	-	-	-	-	-	-	-	-	-	-
3	3	8	(1,3)	8	(1,3)	-	-	-	-	-	-	-	-
4	4	∞	(1,4)	11	(1,2,4)	11	(1,2,4)						
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)						
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)						
P	{1}			{1,2}			{1,2,3}						
T	{2,3,4,5,6}			{3,4,5,6}			{4,5,6}						

- L'algorithme de DIJKSTRA n'a pas détecté le plus court chemin $p = \langle 1, 2, 5, 3 \rangle$ pour aller du sommet 1 vers le sommet 3.
- $l(p) = 5$ alors que L'algorithme de DIJKSTRA nous a retourné $\delta(1, 3) = 8$.
- L'algorithme DIJKSTRA n'est plus utilisable dans le cas d'un graphe valué à poids mixtes.
- D'où la nécessité d'une autre démarche qui prend en considération la présence de poids négatifs, aussi bien des circuits absorbants : Algorithme de FORD BELLMAN.



Plan

- 1 Introduction
- 2 Algorithme de DIJKSTRA
- 3 Algorithme de FORD BELLMAN
 - Principe
 - Application
- 4 Algorithme de FLOYD-WARSHALL
- 5 DIJKSTRA / FORD-BELLMAN / FLOYD-WARSHALL

Principe

Rappel & remarques :

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$ et $\mathcal{C} : A \longrightarrow \mathbb{R}$, la fonction coût associée.

Principe

Rappel & remarques :

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$ et $\mathcal{C} : A \longrightarrow \mathbb{R}$, la fonction coût associée.

- Condition d'optimalité : une condition nécessaire pour avoir un plus court chemin est l'absence de circuits absorbants.

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$ et $\mathcal{C} : A \longrightarrow \mathbb{R}$, la fonction coût associée.

- Condition d'optimalité : une condition nécessaire pour avoir un plus court chemin est l'absence de circuits absorbants.
- Le plus court chemin de la source x_1 à une destination quelconque x_k , $1 \leq k \leq n$, est composé d'au plus n sommets.

Principe

Rappel & remarques :

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$ et $\mathcal{C} : A \longrightarrow \mathbb{R}$, la fonction coût associée.

- Condition d'optimalité : une condition nécessaire pour avoir un plus court chemin est l'absence de circuits absorbants.
- Le plus court chemin de la source x_1 à une destination quelconque x_k , $1 \leq k \leq n$, est composé d'au plus n sommets.
- Tout chemin composé de plus de n sommets passe forcément par un circuit absorbant.

Pseudo-code de l'algorithme de FORD BELLMAN

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$ et $\mathcal{C} : A \rightarrow \mathbb{R}$, la fonction coût associée.

Notation :

- Pour tout $j \in \{1, \dots, n\}$, on note par $dist^k(j)$: la longueur optimale du chemin reliant la source x_1 à la destination x_j , **composé d'au plus k sommets autre que l'origine**. Cette longueur sera mise à jour tout au long du processus afin de déterminer $\delta(x_1, x_j)$, le plus court chemin reliant les deux sommets x_1 et x_j .
- Pour tout $j \in \{1, \dots, n\}$, on note par $Pcc(j)$: le plus court chemin associé au sommet x_j .



Pseudo-code de l'algorithme de FORD-BELLMAN

Pseudo-code

Étape 0 : Initialisation :

- $k = 1$
- $dist^k(1) = 0$
- Pour j de 2 à n faire

$$dist^1(j) = \begin{cases} C_{1,j} & \text{si } (x_1, x_j) \in A, \\ +\infty & \text{sinon.} \end{cases}$$
- fin pour

Étape 1 : Mise à jour des $dist^k(j)$:

- Pour j de 1 à n faire

$$dist^{k+1}(j) = \min(dist^k(j), \min(dist^k(l) + C_{l,j})), \text{ } x_l \text{ sont les prédécesseurs de } x_j.$$
- fin pour

Étape 2 : Test de convergence :

- Si $\forall j \in \{1, \dots, n\}, dist^{k+1}(j) = dist^k(j)$ alors fin : **optimalité atteinte**
- Sinon et Si $k = n - 1$ alors : **il existe un circuit absorbant**
- Sinon, faire $k = k + 1$ et aller à l'étape 1.



Illustration de l'algorithme de FORD BELLMAN : $k = 1$

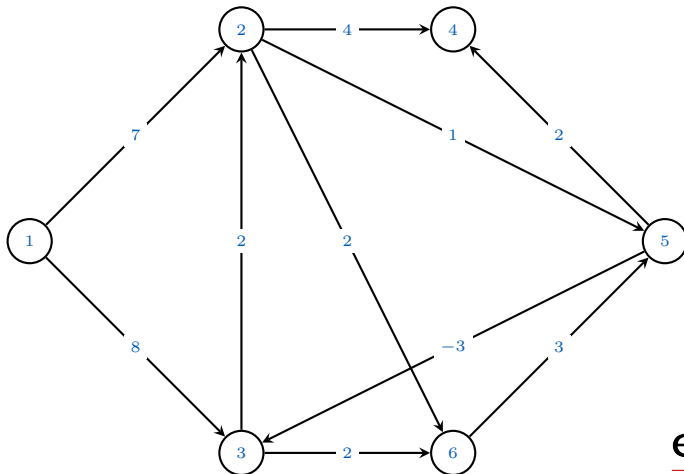


Illustration de l'algorithme de FORD BELLMAN : $k = 1$

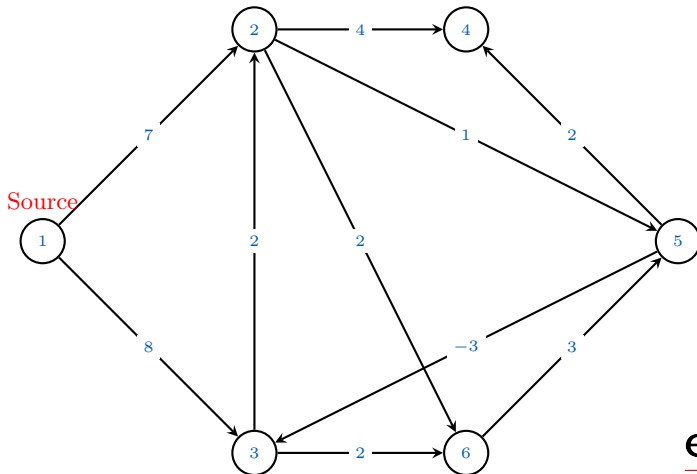


Illustration de l'algorithme de FORD BELLMAN : $k = 1$

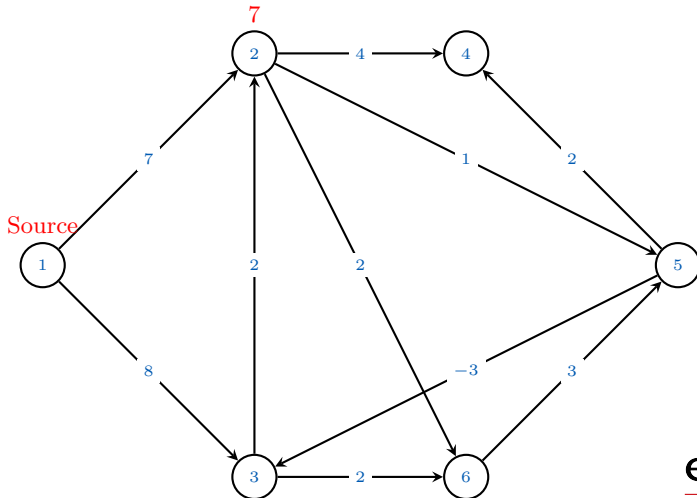


Illustration de l'algorithme de FORD BELLMAN : $k = 1$

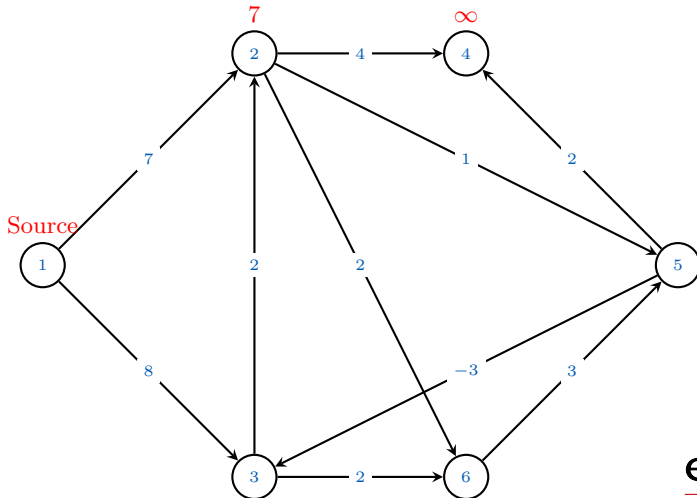


Illustration de l'algorithme de FORD BELLMAN : $k = 1$

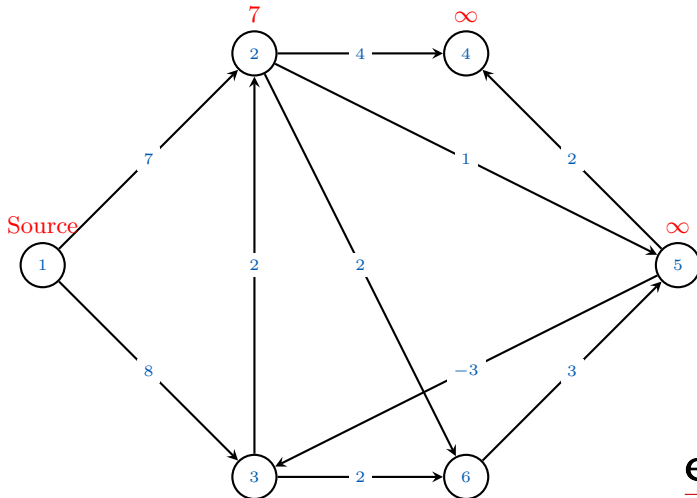




Illustration de l'algorithme de FORD BELLMAN : $k = 1$

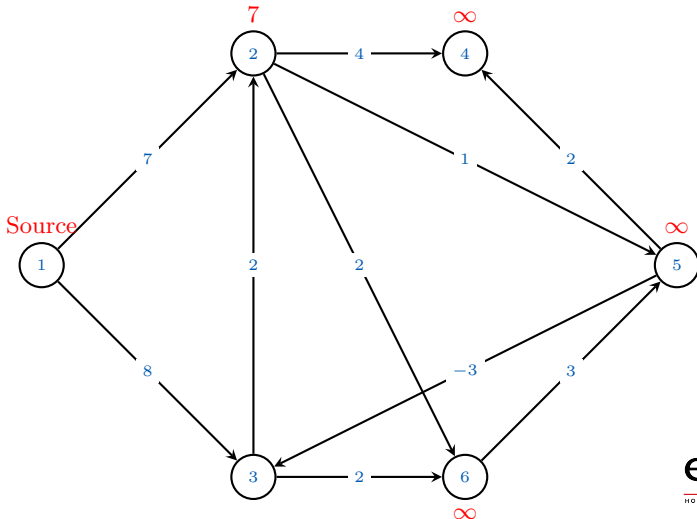
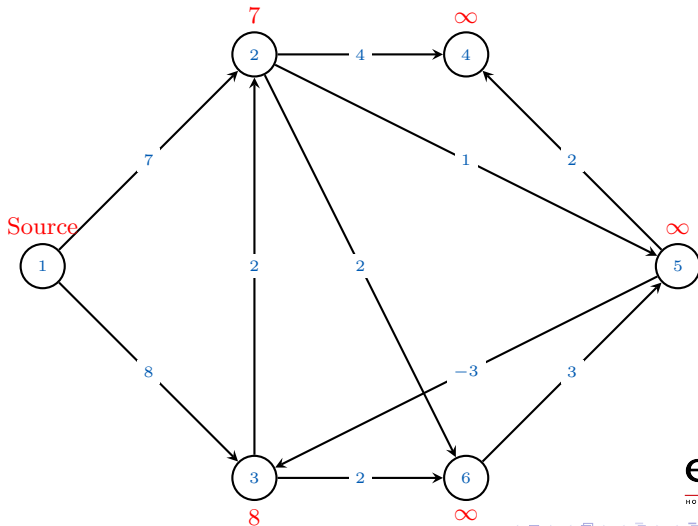


Illustration de l'algorithme de FORD BELLMAN : $k = 1$



Application de l'algorithme de FORD BELLMAN : $k = 1$

$k = 1$			
j	x_j	$dist^1(j)$	$Pcc(j)$
1	1	0	(1,1)
2	2	7	(1,2)
3	3	8	(1,3)
4	4	∞	(1,4)
5	5	∞	(1,5)
6	6	∞	(1,6)

Illustration de l'algorithme de FORD BELLMAN : $k = 2$

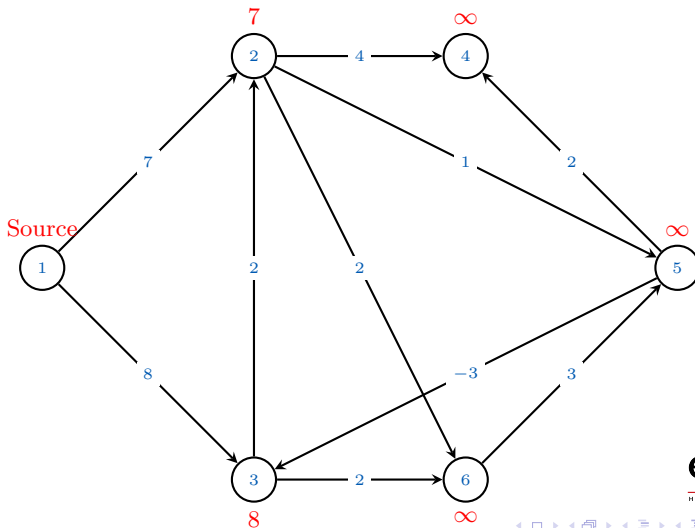
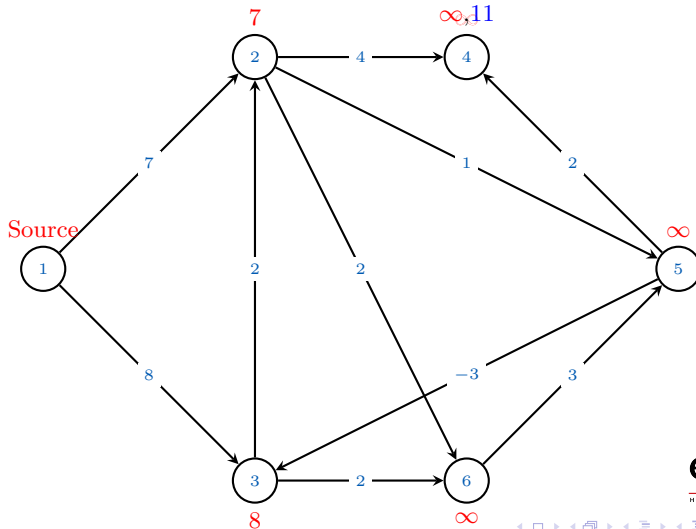


Illustration de l'algorithme de FORD BELLMAN : $k = 2$





Principe

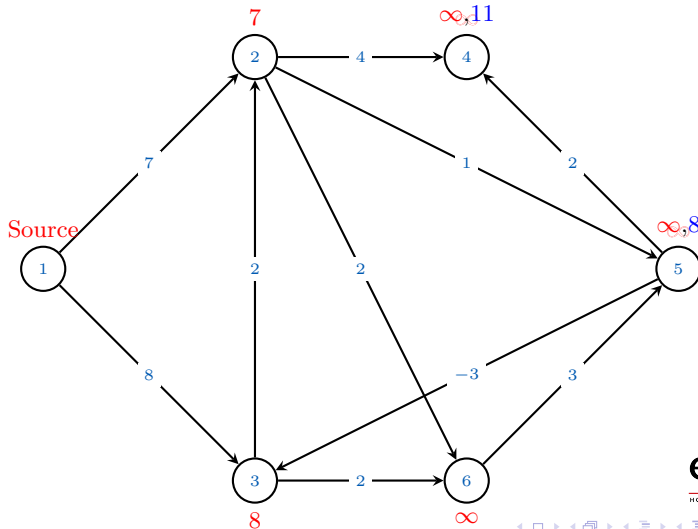
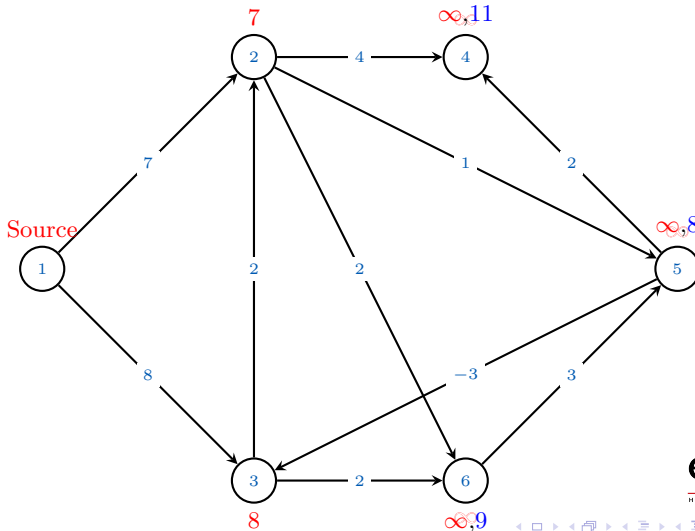
Illustration de l'algorithme de FORD BELLMAN : $k = 2$ 



Illustration de l'algorithme de FORD BELLMAN : $k = 2$





Application de l'algorithme de FORD BELLMAN : $k = 2$



		$k = 1$		$k = 2$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$
1	1	0	(1,1)	0	(1,1)
2	2	7	(1,2)	7	(1,2)
3	3	8	(1,3)	8	(1,3)
4	4	∞	(1,4)	11	(1,2,4)
5	5	∞	(1,5)	8	(1,2,5)
6	6	∞	(1,6)	9	(1,2,6)

Illustration de l'algorithme de FORD BELLMAN : $k = 3$

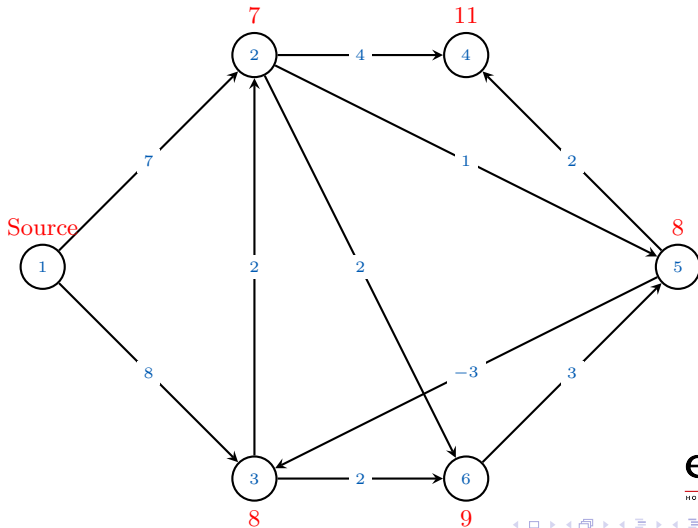
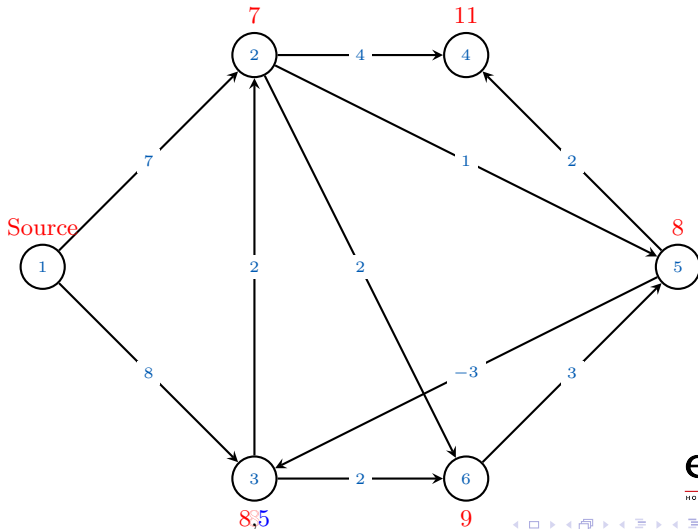


Illustration de l'algorithme de FORD BELLMAN : $k = 3$



Application de l'algorithme de FORD BELLMAN : $k = 3$

j	x_j	$k = 1$		$k = 2$		$k = 3$	
		$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	1	0	(1,1)	0	(1,1)	0	(1,1)
2	2	7	(1,2)	7	(1,2)	7	(1,2)
3	3	8	(1,3)	8	(1,3)	5	(1,2,5,3)
4	4	∞	(1,4)	11	(1,2,4)	10	(1,2,5,4)
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)



Illustration de l'algorithme de FORD BELLMAN : $k = 4$

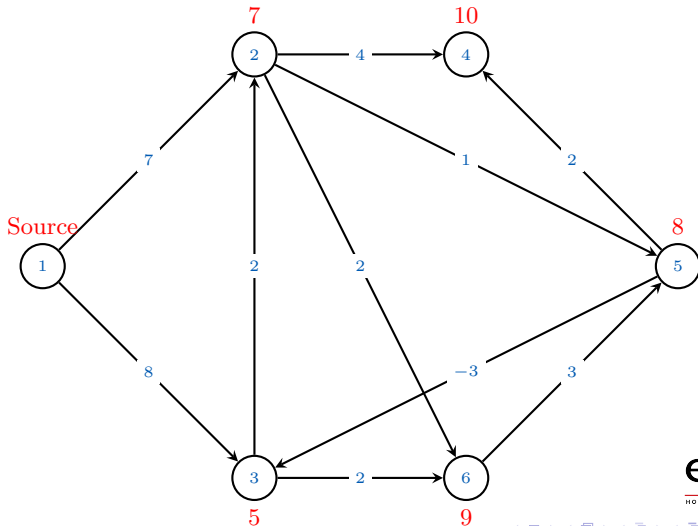
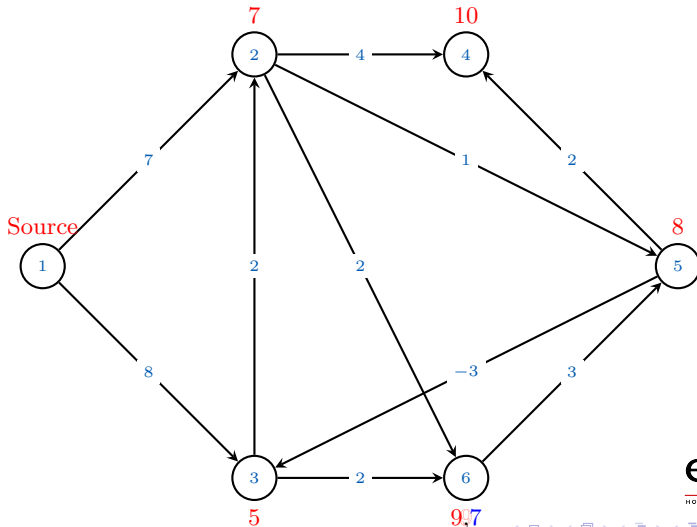
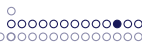


Illustration de l'algorithme de FORD BELLMAN : $k = 4$





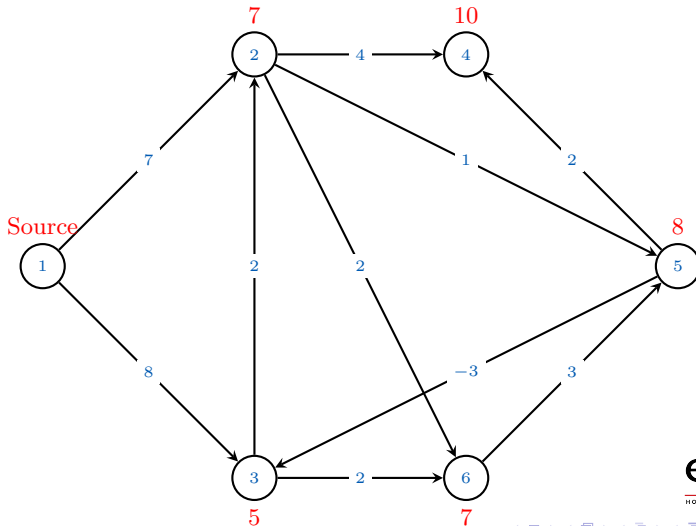
Application de l'algorithme de FORD BELLMAN : $k = 4$



		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	1	0	(1,1)	0	(1,1)	0	(1,1)
2	2	7	(1,2)	7	(1,2)	7	(1,2)
3	3	8	(1,3)	8	(1,3)	5	(1,2,5,3)
4	4	∞	(1,4)	11	(1,2,4)	10	(1,2,5,4)
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)

		$k = 4$	
j	x_j	$dist^4(j)$	$Pcc(j)$
1	1	0	(1,1)
2	2	7	(1,2)
3	3	5	(1,2,5,3)
4	4	10	(1,2,5,4)
5	5	8	(1,2,5)
6	6	7	(1,2,5,3,6)

Illustration de l'algorithme de FORD BELLMAN : $k = 5$





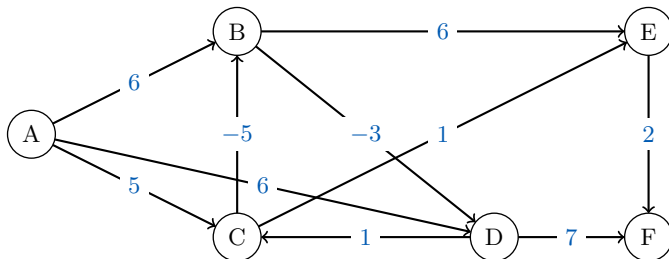
Application de l'algorithme de FORD BELLMAN : $k = 5$



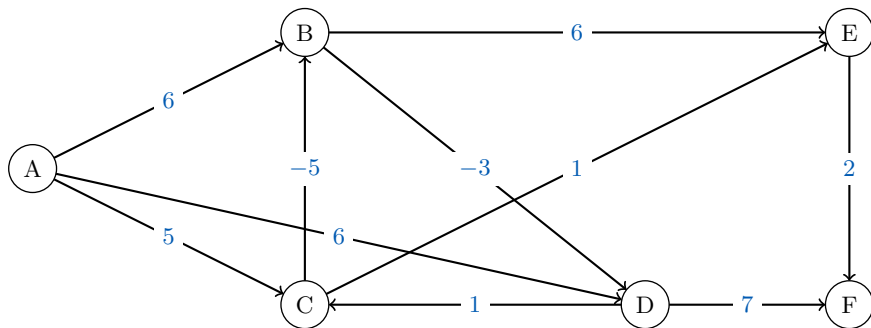
		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	1	0	(1,1)	0	(1,1)	0	(1,1)
2	2	7	(1,2)	7	(1,2)	7	(1,2)
3	3	8	(1,3)	8	(1,3)	5	(1,2,5,3)
4	4	∞	(1,4)	11	(1,2,4)	10	(1,2,5,4)
5	5	∞	(1,5)	8	(1,2,5)	8	(1,2,5)
6	6	∞	(1,6)	9	(1,2,6)	9	(1,2,6)

		$k = 4$		$k = 5$	
j	x_j	$dist^4(j)$	$Pcc(j)$	$dist^5(j)$	$Pcc(j)$
1	1	0	(1,1)	0	(1,1)
2	2	7	(1,2)	7	(1,2)
3	3	5	(1,2,5,3)	5	(1,2,5,3)
4	4	10	(1,2,5,4)	10	(1,2,5,4)
5	5	8	(1,2,5)	8	(1,2,5)
6	6	7	(1,2,5,3,6)	7	(1,2,5,3,6)

Appliquer l'algorithme de FORD BELLMAN pour trouver le chemin de coût minimal entre les sommets A et F du graphe ci-dessous.

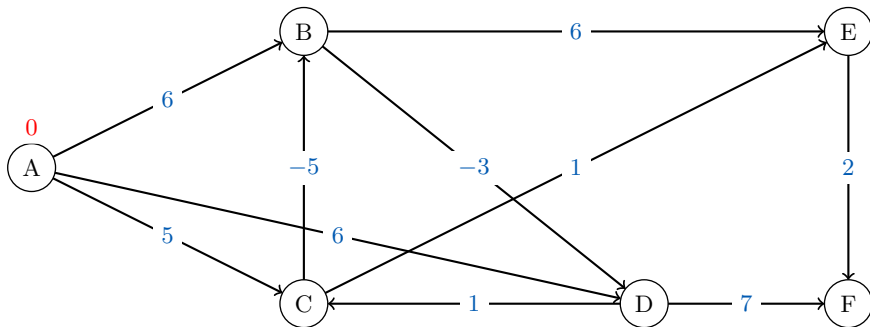


Application de l'algorithme de FORD BELLMAN : $k = 1$

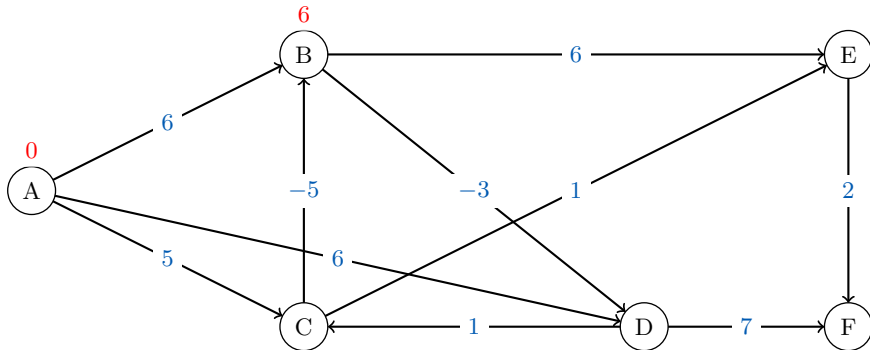


Application

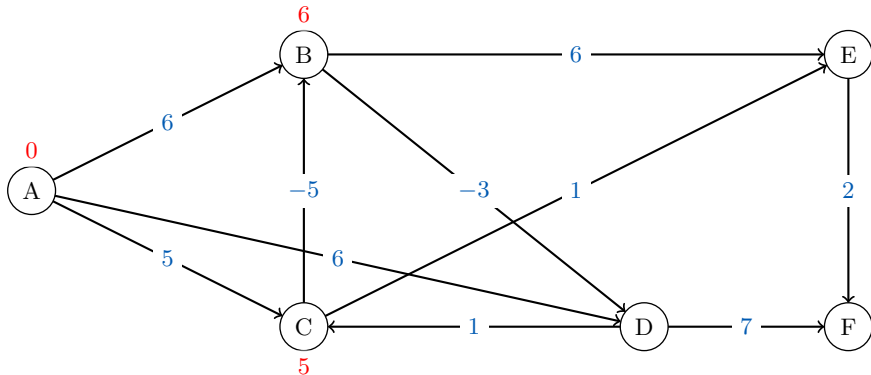
Application de l'algorithme de FORD BELLMAN : $k = 1$



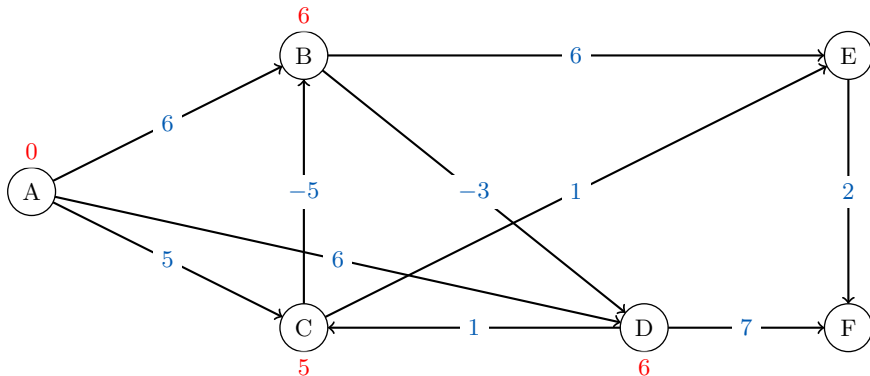
Application de l'algorithme de FORD BELLMAN : $k = 1$



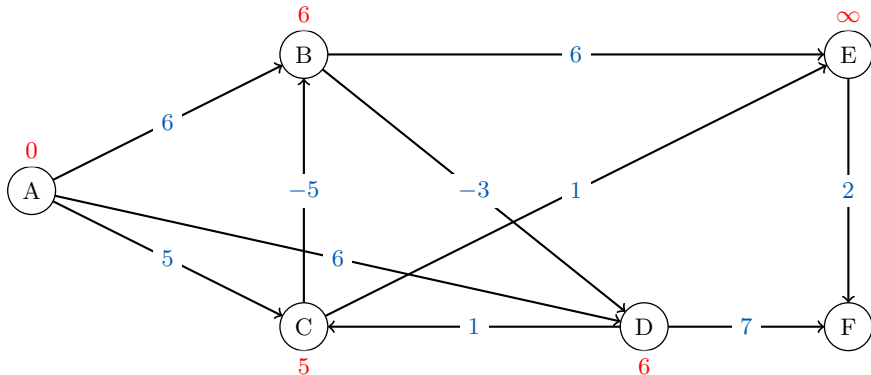
Application de l'algorithme de FORD BELLMAN : $k = 1$



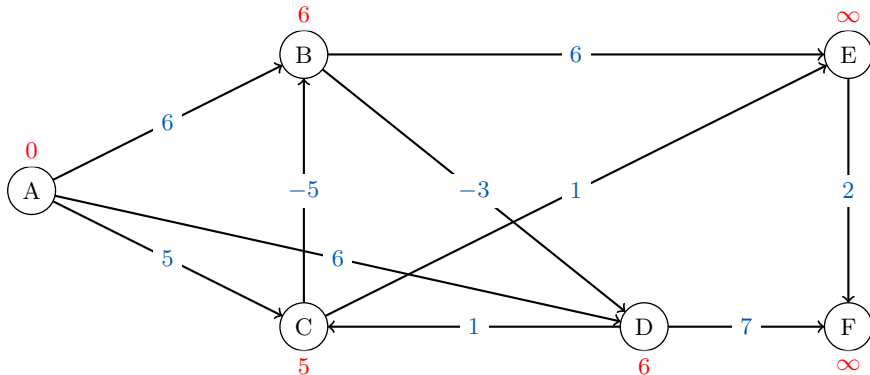
Application de l'algorithme de FORD BELLMAN : $k = 1$



Application de l'algorithme de FORD BELLMAN : $k = 1$



Application de l'algorithme de FORD BELLMAN : $k = 1$



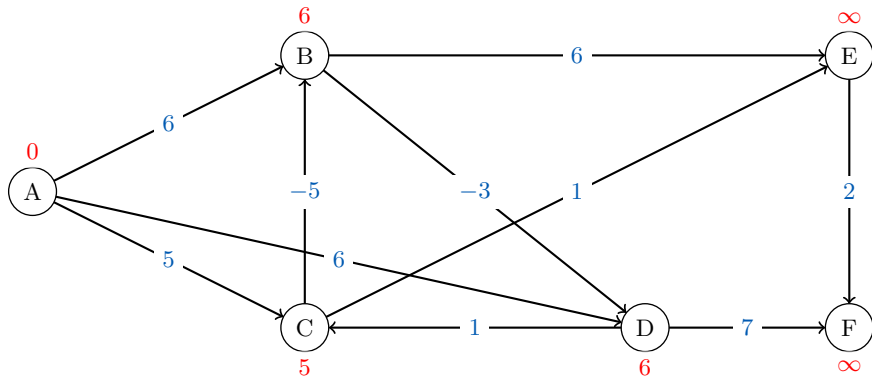


Application de l'algorithme de FORD BELLMAN : $k = 1$

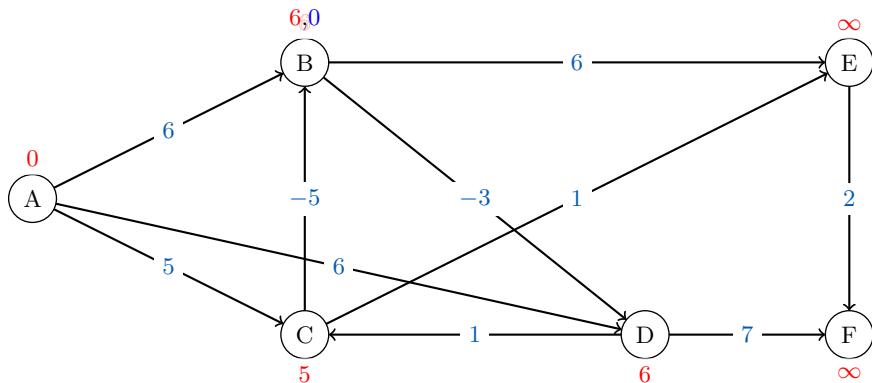


$k = 1$			
j	x_j	$dist^1(j)$	$Pcc(j)$
1	A	0	(A,A)
2	B	6	(A,B)
3	C	5	(A,C)
4	D	6	(A,D)
5	E	∞	(A,E)
6	F	∞	(A,F)

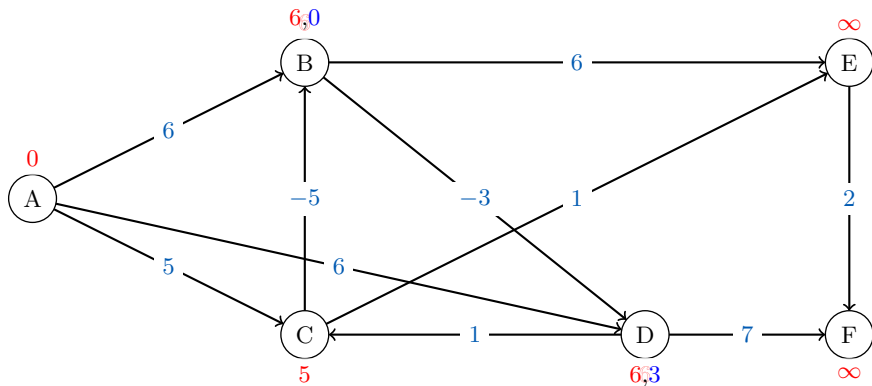
Application de l'algorithme de FORD BELLMAN : $k = 2$



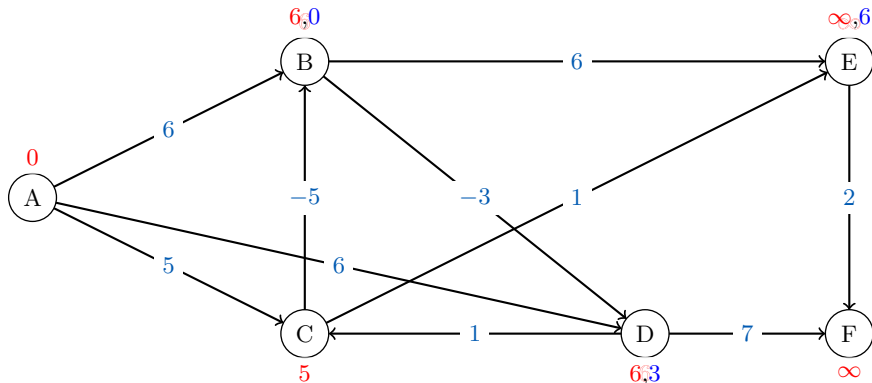
Application de l'algorithme de FORD BELLMAN : $k = 2$



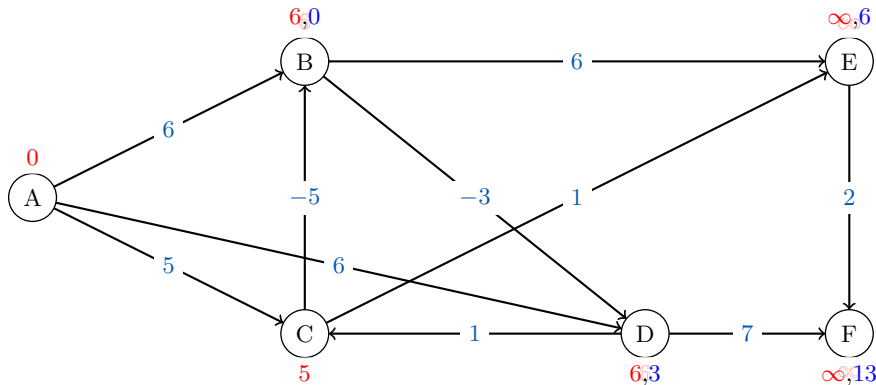
Application de l'algorithme de FORD BELLMAN : $k = 2$



Application de l'algorithme de FORD BELLMAN : $k = 2$



Application de l'algorithme de FORD BELLMAN : $k = 2$



Application de l'algorithme de FORD BELLMAN : $k = 2$

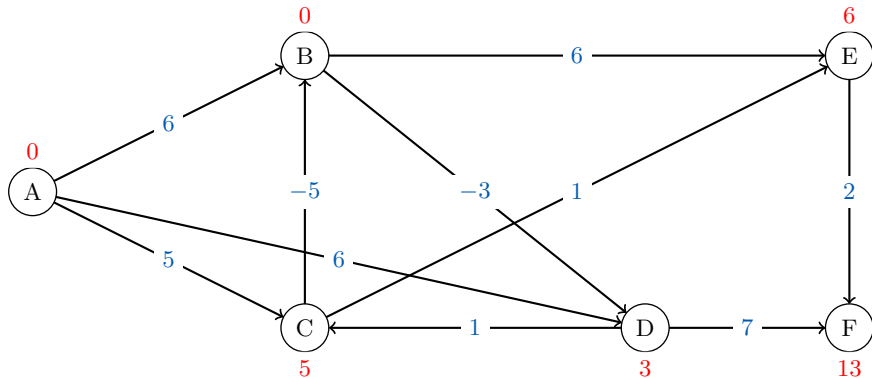
		$k = 1$		$k = 2$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)
4	D	6	(A,D)	3	(A,B,D)
5	E	∞	(A,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)

Application de l'algorithme de FORD BELLMAN : $k = 2$

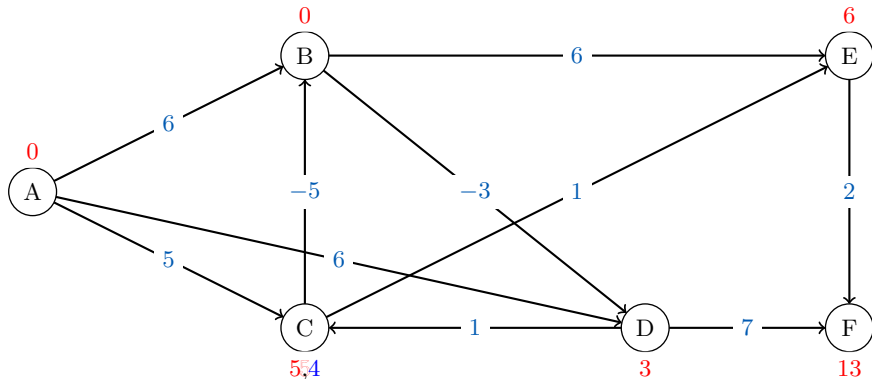
		$k = 1$		$k = 2$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)
4	D	6	(A,D)	3	(A,B,D)
5	E	∞	(A,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)

Test d'arrêt non vérifié : $\exists j \in \{1, \dots, 6\}$ tel que $dist^2(x_j) \neq dist^1(x_j)$

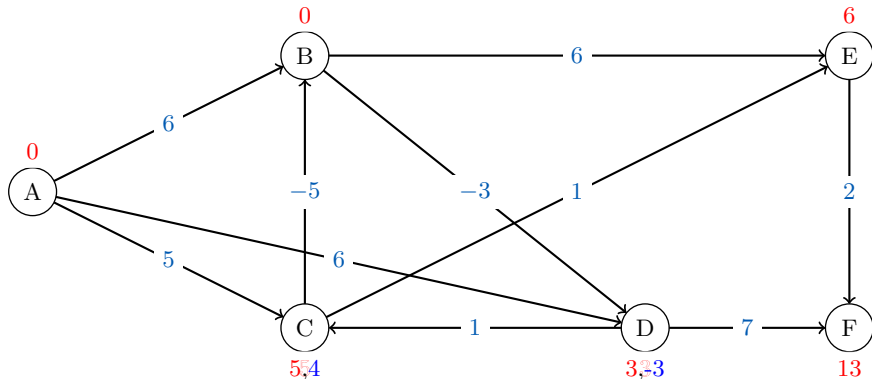
Application de l'algorithme de FORD BELLMAN : $k = 3$



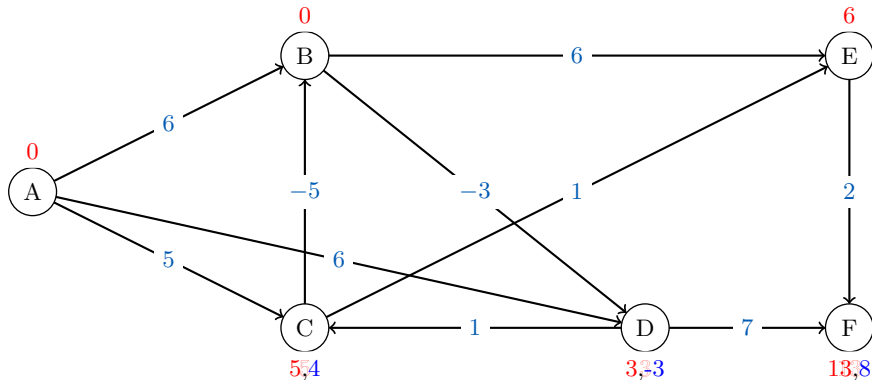
Application de l'algorithme de FORD BELLMAN : $k = 3$



Application de l'algorithme de FORD BELLMAN : $k = 3$



Application de l'algorithme de FORD BELLMAN : $k = 3$





Application de l'algorithme de FORD BELLMAN : $k = 3$

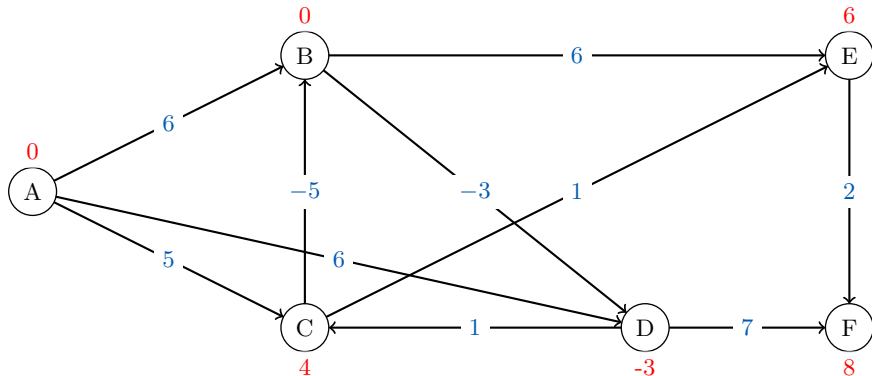
j	x_j	$k = 1$		$k = 2$		$k = 3$	
		$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

Application de l'algorithme de FORD BELLMAN : $k = 3$

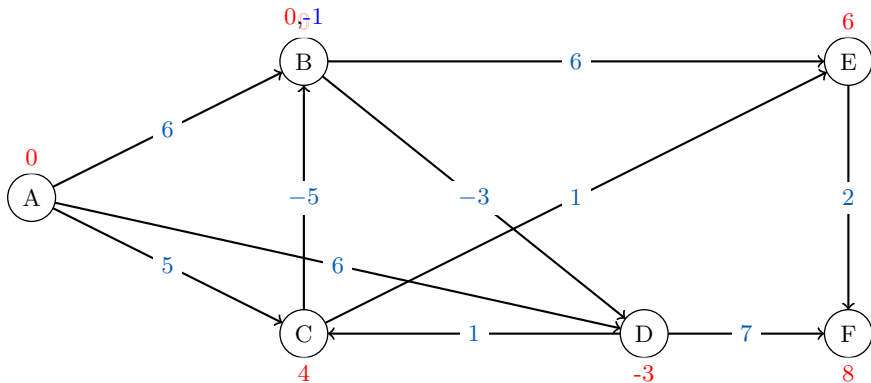
j	x_j	$k = 1$		$k = 2$		$k = 3$	
		$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

Test d'arrêt non vérifié : $\exists j \in \{1, \dots, 6\}$ tel que $dist^3(x_j) \neq dist^2(x_j)$

Application de l'algorithme de FORD BELLMAN : $k = 4$

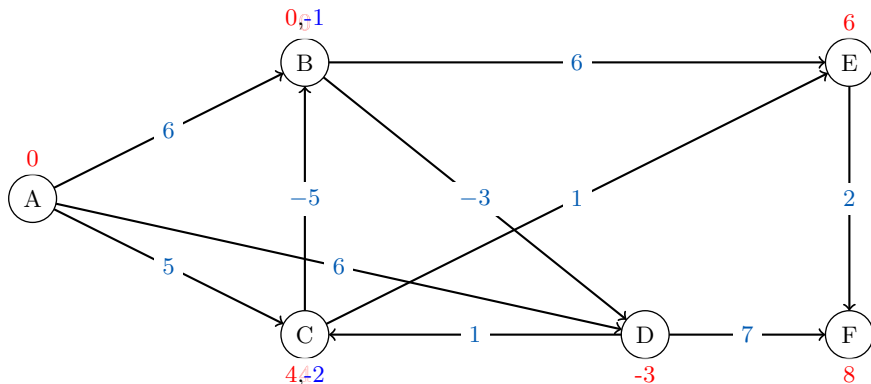


Application de l'algorithme de FORD BELLMAN : $k = 4$

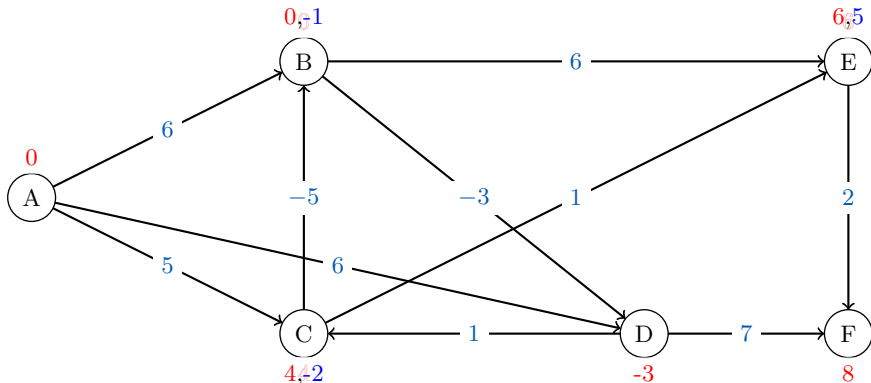




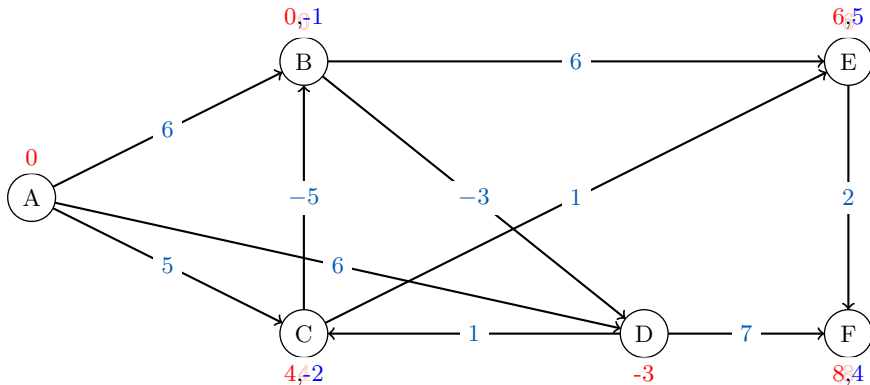
Application de l'algorithme de FORD BELLMAN : $k = 4$



Application de l'algorithme de FORD BELLMAN : $k = 4$



Application de l'algorithme de FORD BELLMAN : $k = 4$



Application de l'algorithme de FORD BELLMAN : $k = 4$

		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

		$k = 4$	
j	x_j	$dist^4(j)$	$Pcc(j)$
1	A	0	(A,A)
2	B	-1	(A,B,D,C,B)
3	C	-2	(A,C,B,D,C)
4	D	-3	(A,C,B,D)
5	E	5	(A,B,D,C,E)
6	F	4	(A,C,B,D,F)



Application de l'algorithme de FORD BELLMAN : $k = 4$

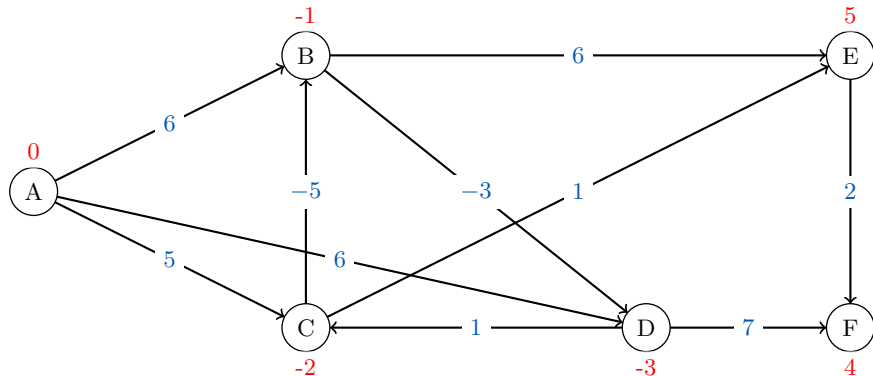
		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

		$k = 4$	
j	x_j	$dist^4(j)$	$Pcc(j)$
1	A	0	(A,A)
2	B	-1	(A,B,D,C,B)
3	C	-2	(A,C,B,D,C)
4	D	-3	(A,C,B,D)
5	E	5	(A,B,D,C,E)
6	F	4	(A,C,B,D,F)

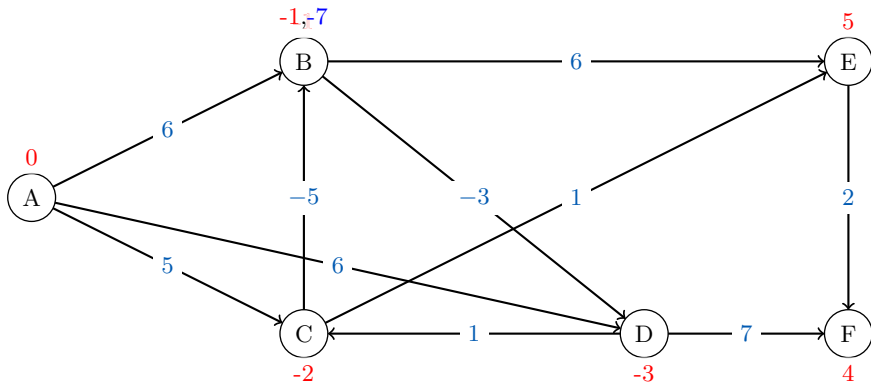
Test d'arrêt non vérifié : $\exists j \in \{1, \dots, 6\}$ tel que $dist^4(x_j) \neq dist^3(x_j)$



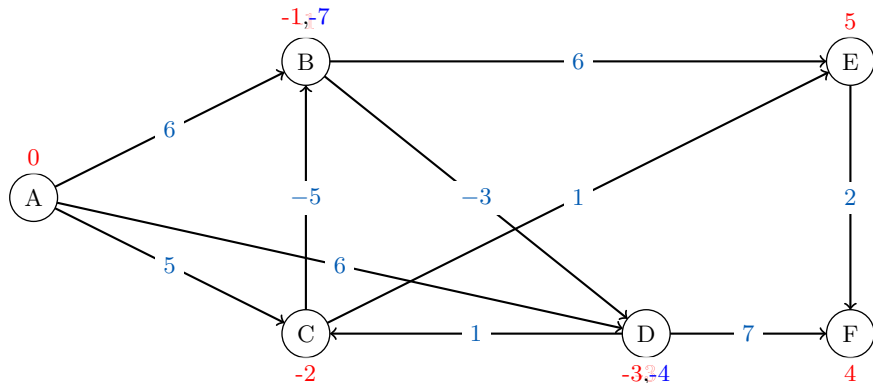
Application de l'algorithme de FORD BELLMAN : $k = 5$



Application de l'algorithme de FORD BELLMAN : $k = 5$



Application de l'algorithme de FORD BELLMAN : $k = 5$





Application de l'algorithme de FORD BELLMAN : $k = 5$



		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

		$k = 4$		$k = 5$	
j	x_j	$dist^4(j)$	$Pcc(j)$	$dist^5(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)
2	B	-1	(A,B,D,C,B)	-7	(A,C,B,D,C,B)
3	C	-2	(A,C,B,D,C)	-2	(A,C,B,D,C)
4	D	-3	(A,C,B,D)	-4	(A,B,D,C,B,D)
5	E	5	(A,B,D,C,E)	-1	(A,C,B,D,C,E)
6	F	4	(A,C,B,D,F)	4	(A,C,B,D,F)



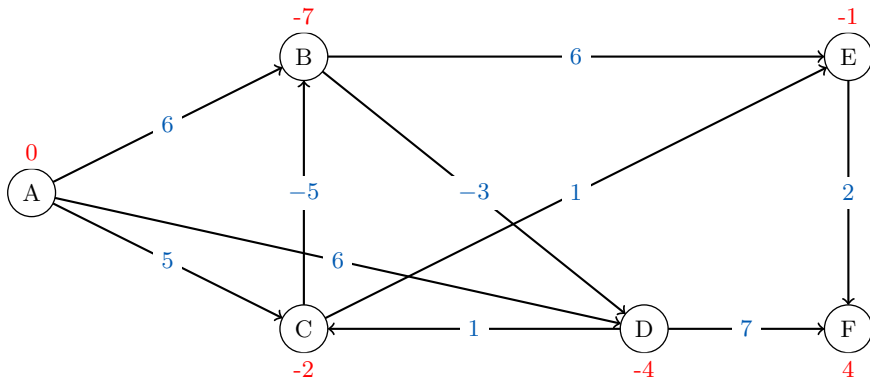
Application de l'algorithme de FORD BELLMAN : $k = 5$

		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

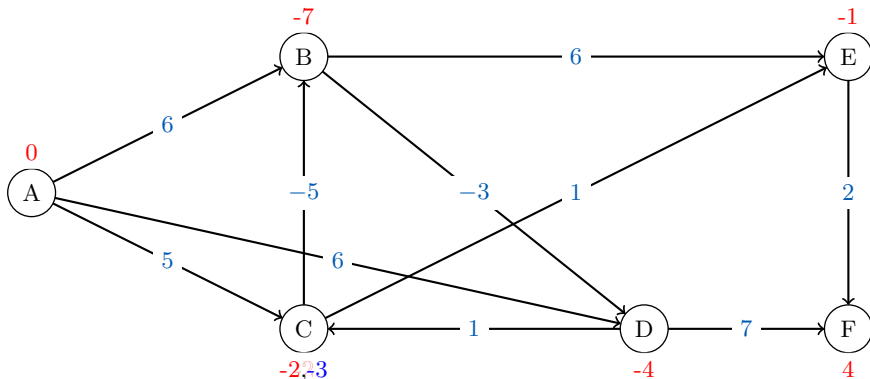
		$k = 4$		$k = 5$	
j	x_j	$dist^4(j)$	$Pcc(j)$	$dist^5(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)
2	B	-1	(A,B,D,C,B)	-7	(A,C,B,D,C,B)
3	C	-2	(A,C,B,D,C)	-2	(A,C,B,D,C)
4	D	-3	(A,C,B,D)	-4	(A,B,D,C,B,D)
5	E	5	(A,B,D,C,E)	-1	(A,C,B,D,C,E)
6	F	4	(A,C,B,D,F)	4	(A,C,B,D,F)

Test d'arrêt non vérifié : $\exists j \in \{1, \dots, 6\}$ tel que $dist^5(x_j) \neq dist^4(x_j)$

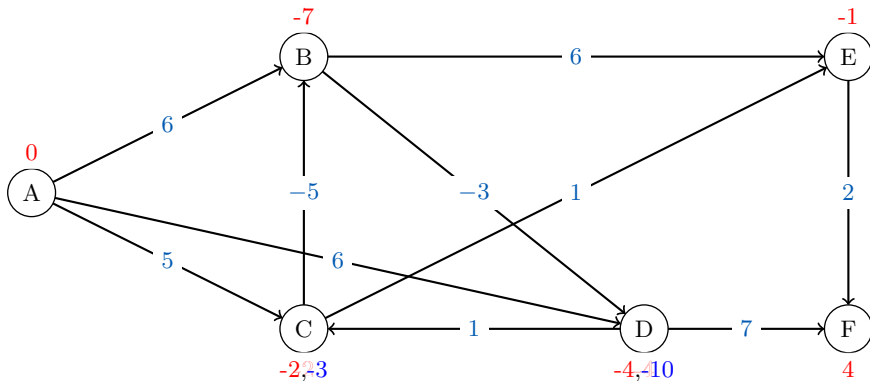
Application de l'algorithme de FORD BELLMAN : $k = 6$



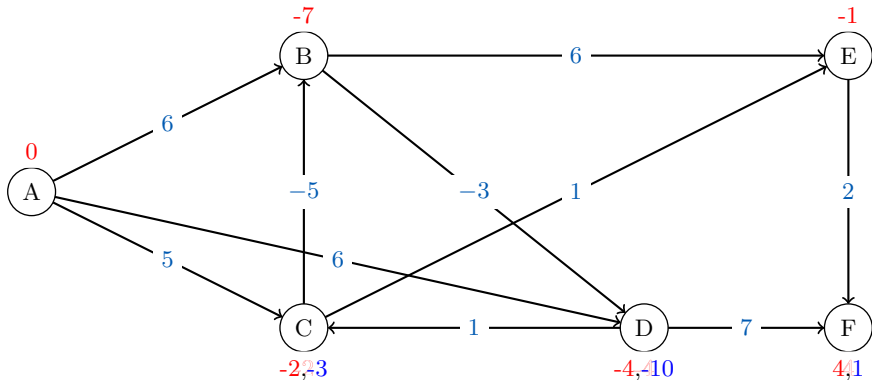
Application de l'algorithme de FORD BELLMAN : $k = 6$



Application de l'algorithme de FORD BELLMAN : $k = 6$



Application

Application de l'algorithme de FORD BELLMAN : $k = 6$ 



Application de l'algorithme de FORD BELLMAN : $k = 6$



		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

		$k = 4$		$k = 5$		$k = 6$	
j	x_j	$dist^4(j)$	$Pcc(j)$	$dist^5(j)$	$Pcc(j)$	$dist^6(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	-1	(A,B,D,C,B)	-7	(A,C,B,D,C,B)	-7	(A,C,B,D,C,B)
3	C	-2	(A,C,B,D,C)	-2	(A,C,B,D,C)	-3	(A,B,D,C,B,D,C)
4	D	-3	(A,C,B,D)	-4	(A,B,D,C,B,D)	-10	(A,C,B,D,C,B,D)
5	E	5	(A,B,D,C,E)	-1	(A,C,B,D,C,E)	-1	(A,C,B,D,C,E)
6	F	4	(A,C,B,D,F)	4	(A,C,B,D,F)	1	(A,B,D,C,B,D,F)



Application de l'algorithme de FORD BELLMAN : $k = 6$

		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

		$k = 4$		$k = 5$		$k = 6$	
j	x_j	$dist^4(j)$	$Pcc(j)$	$dist^5(j)$	$Pcc(j)$	$dist^6(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	-1	(A,B,D,C,B)	-7	(A,C,B,D,C,B)	-7	(A,C,B,D,C,B)
3	C	-2	(A,C,B,D,C)	-2	(A,C,B,D,C)	-3	(A,B,D,C,B,D,C)
4	D	-3	(A,C,B,D)	-4	(A,B,D,C,B,D)	-10	(A,C,B,D,C,B,D)
5	E	5	(A,B,D,C,E)	-1	(A,C,B,D,C,E)	-1	(A,C,B,D,C,E)
6	F	4	(A,C,B,D,F)	4	(A,C,B,D,F)	1	(A,B,D,C,B,D,F)

Test d'arrêt vérifié : $\exists j \in \{1, \dots, 6\}$ tel que $dist^6(x_j) \neq dist^5(x_j)$

& $k = n$:



Application de l'algorithme de FORD BELLMAN : $k = 6$

		$k = 1$		$k = 2$		$k = 3$	
j	x_j	$dist^1(j)$	$Pcc(j)$	$dist^2(j)$	$Pcc(j)$	$dist^3(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	6	(A,B)	0	(A,C,B)	0	(A,C,B)
3	C	5	(A,C)	5	(A,C)	4	(A,B,D,C)
4	D	6	(A,D)	3	(A,B,D)	-3	(A,C,B,D)
5	E	∞	(A,E)	6	(A,C,E)	6	(A,C,E)
6	F	∞	(A,F)	13	(A,D,F)	8	(A,B,E,F)

		$k = 4$		$k = 5$		$k = 6$	
j	x_j	$dist^4(j)$	$Pcc(j)$	$dist^5(j)$	$Pcc(j)$	$dist^6(j)$	$Pcc(j)$
1	A	0	(A,A)	0	(A,A)	0	(A,A)
2	B	-1	(A,B,D,C,B)	-7	(A,C,B,D,C,B)	-7	(A,C,B,D,C,B)
3	C	-2	(A,C,B,D,C)	-2	(A,C,B,D,C)	-3	(A,B,D,C,B,D,C)
4	D	-3	(A,C,B,D)	-4	(A,B,D,C,B,D)	-10	(A,C,B,D,C,B,D)
5	E	5	(A,B,D,C,E)	-1	(A,C,B,D,C,E)	-1	(A,C,B,D,C,E)
6	F	4	(A,C,B,D,F)	4	(A,C,B,D,F)	1	(A,B,D,C,B,D,F)

Test d'arrêt vérifié : $\exists j \in \{1, \dots, 6\}$ tel que $dist^6(x_j) \neq dist^5(x_j)$
 & $k = n$: existence de circuit absorbant.



Plan

1 Introduction

2 Algorithme de DIJKSTRA

3 Algorithme de FORD BELLMAN

4 Algorithme de FLOYD-WARSHALL

- Principe
- Application

5 DIJKSTRA / FORD-BELLMAN / FLOYD-WARSHALL

Principe

Pseudo-code de l'algorithme de FLOYD-WARSHALL

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$ et $\mathcal{C} : A \longrightarrow \mathbb{R}$, la fonction coût associée.

Notation :

- Pour tout $i, j \in \{1, \dots, n\}$, on note par $dist(i, j)$: la longueur optimale du chemin reliant la source x_i à la destination x_j . Cette longueur sera mise à jour tout au long du processus afin de déterminer $\delta(x_i, x_j)$, le plus court chemin reliant les deux sommets x_i et x_j .
- Pour tout $i, j \in \{1, \dots, n\}$, on note par $Pcc(i, j)$: le plus court chemin reliant les sommets x_i et x_j .



Principe

Pseudo-code de l'algorithme de FLOYD-WARSHALL

Pseudo-code

Étape 0 : Initialisation :

- Pour j de 1 à n faire

$$\text{dist}(i, j) = \begin{cases} 0 & \text{si } i = j, \\ C_{i,j} & \text{si } (x_i, x_j) \in A, \\ +\infty & \text{sinon.} \end{cases}$$

- fin si

$$\forall i \neq j, P_{cc}(i, j) = (x_i, x_j)$$

Étape 1 : Mise à jour des $\text{dist}(i, j)$:

- $k = 1$
- $\forall i \neq k$ et $j \neq k$, si $\text{dist}(i, k) + \text{dist}(k, j) < \text{dist}(i, j)$ faire
 $\text{dist}(i, j) = \text{dist}(i, k) + \text{dist}(k, j)$, et $P_{cc}(i, j) = P_{cc}(i, k) \cup P_{cc}(k, j)$.
- fin pour

Étape 2 : Test de convergence :

- Si $\exists i$ tel que $\text{dist}(i, i) < 0$ alors fin : il existe un circuit absorbant
- Sinon et Si $k = n$ alors : optimalité atteinte
- Sinon, faire $k = k + 1$ et aller à l'étape 1.



Principe

Illustration de l'algorithme de FLOYD-WARSHALL

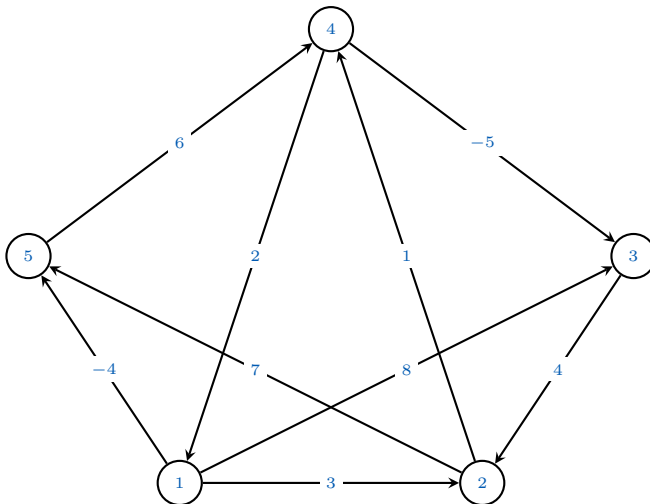
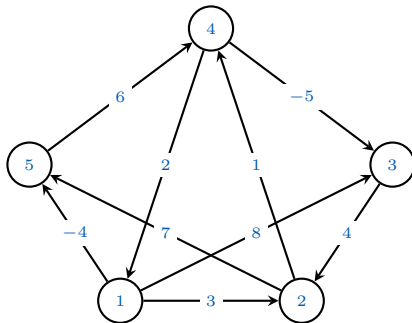


Illustration de l'algorithme de FLOYD-WARSHALL / initialisation



$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(0)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & X & 4 & X & X \\ X & X & X & 5 & \end{pmatrix}$$



Illustration de l'algorithme de FLOYD-WARSHALL / $k = 1$

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(0)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & X & 4 & X & X \\ X & X & X & 5 & X \end{pmatrix}$$



Illustration de l'algorithme de FLOYD-WARSHALL / $k = 1$

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(0)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & X & 4 & X & X \\ X & X & X & 5 & X \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & & & & \\ \infty & & & & \\ 2 & & & & \\ \infty & & & & \end{pmatrix}$$

$$\pi^{(1)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & & & & \\ X & & & & \\ 4 & & & & \\ X & & & & \end{pmatrix}$$





Illustration de l'algorithme de FLOYD-WARSHALL / $k = 1$

$$D^{(0)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$



$$\pi^{(0)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & X & 4 & X & X \\ X & X & X & 5 & X \end{pmatrix}$$

$$\pi^{(1)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

Illustration de l'algorithme de FLOYD-WARSHALL / $k = 2$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(1)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$



Illustration de l'algorithme de FLOYD-WARSHALL / $k = 2$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(1)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} & 3 & & & \\ \infty & 0 & \infty & 1 & 7 \\ & 4 & & & \\ & 5 & & & \\ & \infty & & & \end{pmatrix}$$

$$\pi^{(2)} = \begin{pmatrix} & 1 & & & \\ X & X & X & 2 & 2 \\ & 3 & & & \\ & 1 & & & \\ & X & & & \end{pmatrix}$$



Principe

Illustration de l'algorithme de FLOYD-WARSHALL / $k = 2$

$$D^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(1)} = \begin{pmatrix} X & 1 & 1 & X & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & X & X \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(2)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & 2 & 2 \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

Principe

Illustration de l'algorithme de FLOYD-WARSHALL / $k = 3$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(2)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & 2 & 2 \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$



Illustration de l'algorithme de FLOYD-WARSHALL / $k = 3$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(2)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & 2 & 2 \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} & & 8 & & \\ & & \infty & & \\ \infty & 4 & 0 & 5 & 11 \\ & & -5 & & \\ & & \infty & & \end{pmatrix}$$

$$\pi^{(3)} = \begin{pmatrix} & & 1 & & \\ & & X & & \\ X & 3 & X & 2 & 2 \\ & & 4 & & \\ & & X & & \end{pmatrix}$$





Illustration de l'algorithme de FLOYD-WARSHALL / $k = 3$

$$D^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(2)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & 2 & 2 \\ 4 & 1 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ 8 & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(3)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ 8 & 3 & X & 2 & 2 \\ 4 & 3 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$





Illustration de l'algorithme de FLOYD-WARSHALL / $k = 4$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(3)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & 2 & 2 \\ 4 & 3 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$



Principe

Illustration de l'algorithme de FLOYD-WARSHALL / $k = 4$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$\pi^{(3)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & 2 & 2 \\ 4 & 3 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} & & & 4 \\ & & & 1 \\ & & & 5 \\ 2 & -1 & -5 & 0 & -2 \\ & & & 6 \end{pmatrix}$$

$$\pi^{(4)} = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 4 & 3 & 4 & X & 1 \\ 5 \end{pmatrix}$$



Illustration de l'algorithme de FLOYD-WARSHALL / $k = 4$

$$D^{(3)} = \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$\pi^{(3)} = \begin{pmatrix} X & 1 & 1 & 2 & 1 \\ X & X & X & 2 & 2 \\ X & 3 & X & 2 & 2 \\ 4 & 3 & 4 & X & 1 \\ X & X & X & 5 & X \end{pmatrix}$$

$$\pi^{(4)} = \begin{pmatrix} X & 1 & 4 & 2 & 1 \\ 4 & X & 4 & 2 & 4 \\ 4 & 3 & X & 2 & 4 \\ 4 & 3 & 4 & X & 1 \\ 4 & 4 & 4 & 5 & X \end{pmatrix}$$



Principe

Illustration de l'algorithme de FLOYD-WARSHALL / $k = 5$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$\pi^{(4)} = \begin{pmatrix} X & 1 & 4 & 2 & 1 \\ 4 & X & 4 & 2 & 4 \\ 4 & 3 & X & 2 & 4 \\ 4 & 3 & 4 & X & 1 \\ 4 & 4 & 4 & 5 & X \end{pmatrix}$$

Principe

Illustration de l'algorithme de FLOYD-WARSHALL / $k = 5$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$\pi^{(4)} = \begin{pmatrix} X & 1 & 4 & 2 & 1 \\ 4 & X & 4 & 2 & 4 \\ 4 & 3 & X & 2 & 4 \\ 4 & 3 & 4 & X & 1 \\ 4 & 4 & 4 & 5 & X \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} & & & & -4 \\ & & & & -1 \\ & & & & 3 \\ & & & & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$\pi^{(5)} = \begin{pmatrix} & & & & 1 \\ & & & & 4 \\ & & & & 4 \\ & & & & 1 \\ 4 & 4 & 4 & 5 & X \end{pmatrix}$$



Illustration de l'algorithme de FLOYD-WARSHALL / $k = 5$

$$D^{(4)} = \begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

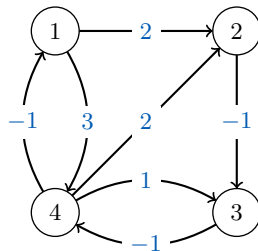
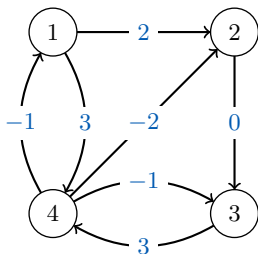


$$\pi^{(4)} = \begin{pmatrix} X & 1 & 4 & 2 & 1 \\ 4 & X & 4 & 2 & 4 \\ 4 & 3 & X & 2 & 4 \\ 4 & 3 & 4 & X & 1 \\ 4 & 4 & 4 & 5 & X \end{pmatrix}$$

$$\pi^{(5)} = \begin{pmatrix} X & 5 & 5 & 5 & 1 \\ 4 & X & 4 & 2 & 4 \\ 4 & 3 & X & 2 & 4 \\ 4 & 3 & 4 & X & 1 \\ 4 & 4 & 4 & 5 & X \end{pmatrix}$$

Application

Appliquer l'algorithme de Floyd sur les deux graphes suivants.





Plan

- 1 Introduction
- 2 Algorithme de DIJKSTRA
- 3 Algorithme de FORD BELLMAN
- 4 Algorithme de FLOYD-WARSHALL
- 5 DIJKSTRA / FORD-BELLMAN / FLOYD-WARSHALL



DIJKSTRA / FORD-BELLMAN / FLOYD-WARSHALL

On considère un graphe orienté valué $G = (X, A)$ avec $|X| = n$, $|A| = m$ et $\mathcal{C} : A \rightarrow \mathbb{R}$, la fonction coût associée.

	DIJKSTRA	FORD BELLMAN	FLOYD WARSHALL
Type de chemins	d'un sommet vers les autres	d'un sommet vers les autres	paires de sommets
Complexité représentation du graphe par listes d'adjacence	$\theta(n + m \log(n))$	$\theta(n \times m)$	
Complexité représentation du graphe par matrice d'adjacence	$\theta(n^2)$	$\theta(n^3)$	$\theta(n^3)$
Cas d'utilisation	Graphes à poids positifs	Graphes à poids quelconques	Graphes à poids quelconques