# Plus court chemin

Christophe Viroulaud

Terminale - NSI

Archi 15

### Plus court chemin

Représentation des réseaux

Graphe non oriente

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme de Dijkstra

Principe

Pour calculer rapidement les distances dans le réseau, les routeurs appliquent des algorithmes conçus au début de l'ère de l'informatique.

Comment fonctionnent les algorithmes de plus court chemin?

### Plus court chemin

Représentation des éseaux

Graphe non orienté

Protocole RIP

Principe

Aise en application

OSPF : algorithme de Diikstra

Principe

Mise en applicatio Complexité

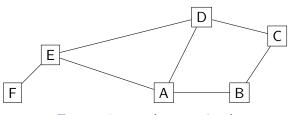
### Sommaire

### Plus court chemin

- 1. Représentation des réseaux

### Représentation des réseaux

# Représentation des réseaux - Graphe non orienté



 $FIGURE\ 1-graphe\ non\ orienté$ 

# À retenir

Un graphe est composé :

- ► de sommets ou nœuds,
- d'arêtes ou arcs qui relient les sommets.

### Plus court chemin

eprésentation des seaux

### Graphe non orienté

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithme de Dijkstra

Principe Mise en application Complexité

# F A B

FIGURE 2 – graphe non orienté

# À retenir

Pour chaque nœud on peut définir ses **prédécesseurs**. **E** est le prédécesseur de **F**.

### Plus court chemin

Représentation des réseaux

### Graphe non orienté

Graphe pondéré

Protocole KIP : algorithme de Bellman-Ford

Principe

iviise en applic

# OSPF : algorithme de Dijkstra

Principe

# Sommaire

- 1. Représentation des réseaux
- 1.1 Graphe non orienté
- 1.2 Graphe pondéré
- 2. Protocole RIP: algorithme de Bellman-Ford
- 3. OSPF : algorithme de Diikstra

### Plus court chemin

Représentation des réseaux

raphe non orienté

### Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en appl

OSPF: algorithme

Principe

# Graphe pondéré

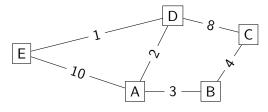


FIGURE 3 – graphe pondéré

# À retenir

Selon le protocole mis en place la pondération pourra représenter :

- RIP : le nombre de réseaux traversés,
- ► OSPF : le coût d'un réseau.

### Plus court chemin

Représentation des réseaux

Graphe non orien

### Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

-rincipe Vlise en annlic

omplexité

### OSPF : algorithme de Dijkstra

# Sommaire

Plus court chemin

- 1. Représentation des réseaux
- 2. Protocole RIP: algorithme de Bellman-Ford
- 2.1 Principe
- 2.2 Mise en application
- 2.3 Complexité
- 3. OSPF: algorithme de Diikstra

Représentation des réseaux

Granhe non orienté

Graphe pond

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en applicatio

OSPF: algorithme

Principe

**1956 - 1958** 

### Plus court chemin

Représentation des éseaux

Graphe non orienté

Protocole RIP : algorithme de Bellman-Ford

### Principe

Mise en application

# OSPF : algorithme

- **1956 1958**
- Richard Bellman (père programmation dynamique)

### Plus court chemin

eprésentation des

Graphe non orienté

Protocole RIP algorithme de Bellman-Ford

### Principe

Mise en application

OSPF: algorithme

- **1956 1958**
- Richard Bellman (père programmation dynamique)
- Lester Ford (problème de flot maximum)

### Plus court chemin

Graphe non orienté

Protocole RIP algorithme de Bellman-Ford

### Principe

Mise en application

OSPF: algorithme

- 1956 1958
- Richard Bellman (père programmation dynamique)
- Lester Ford (problème de flot maximum)
- redécouvert par Edward Moore en 1959

### Plus court chemin

### Principe

- 1956 1958
- Richard Bellman (père programmation dynamique)
- Lester Ford (problème de flot maximum)
- redécouvert par Edward Moore en 1959
- Le protocole RIP applique cet algorithme.

### Plus court chemin

#### Principe

# Remarque

L'algorithme de Bellman-Ford est normalement appliqué dans un graphe orienté. Cependant, si les pondérations sont toutes positives, il est possible d'utiliser un graphe non orienté.

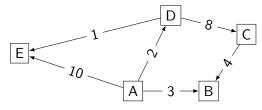


FIGURE 4 – graphe orienté

### Plus court chemin

Représentation des

raphe non orient

Graphe pondéré

Protocole RIP

### Principe

Mise en application

OSPF : algorithme de Dijkstra

### Principe

# À retenir

La distance pour atteindre chaque nœud correspond à la distance pour atteindre son prédécesseur à laquelle on ajoute le poids de l'arête les séparant.

### Plus court chemin

Représentation des éseaux

Graphe non orienté

Protocole RIP Ilgorithme de Bellman-Ford

#### Principe

Mise en application

Complexité

# OSPF : algorithme de Dijkstra

### Principe

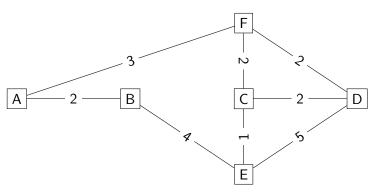


FIGURE 5 – graphe orienté pondéré

# À retenir

L'algorithme de Bellman-Ford applique une méthode récursive.

### Plus court chemin

Représentation des réseaux

Graphe nondéré

Protocole RIP : algorithme de

### Principe

Mise en applica

OSPF : algorithme

Principe

Mise en applicatio

Complexité

Pour chaque routeur, on obtient un tableau contenant la

distance minimale entre le routeur de départ et chaque autre

routeur.

### Plus court chemin

Représentation des réseaux

Graphe non orienté

Protocole RIP algorithme de

### Principe

Mise en application

# OSPF : algorithme de Diikstra

# Sommaire

- 1. Representation des reseaux
- 2. Protocole RIP: algorithme de Bellman-Ford
- 2.1 Principe
- 2.2 Mise en application
- 2.3 Complexité
- 3. OSPF: algorithme de Dijkstra

### Plus court chemin

Représentation des éseaux

Graphe non orienté

Protocole RIP algorithme de

Dringing

Mise en application

OSPF : algorithme

Principe

Mise en application Complexité

# Mise en application

### Initialisation:

- Créer un tableau des distances entre A et les routeurs. initialisées à l'infini.
- Modifier la distance vers A à 0.

### Plus court chemin

- Créer un tableau des distances entre A et les routeurs, initialisées à l'infini.
- ► Modifier la distance vers A à 0.

### Déroulement :

- ► Tant que (nombre d'itérations) < (nombre de routeurs)
  - Pour chaque arc du graphe
    - Si (distance du routeur) > (distance de son prédécesseur + poids de l'arc entre les deux routeurs)
       ⇒ Mettre à jour distance du routeur

Représentation des éseaux

Fraphe non orienté

Protocole RIP algorithme de Bellman-Ford

incipe

Mise en application

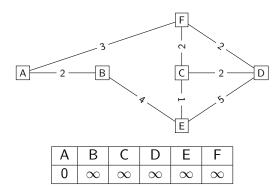
OSPF : algorith

# **Observations**

- On effectue autant d'itérations qu'il y a de routeurs.
- On regarde chaque arc à chaque tour.

### Plus court chemin

# Initialisation



### Plus court chemin

Représentation des

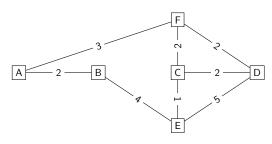
Graphe nondéré

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme



Α	В	C	D	Е	F
0	2 (A)	$\infty$	$\infty$	$\infty$	$\infty$

La distance calculée pour atteindre B correspond à la distance de son prédécesseur (A) augmentée du poids de l'arc A-B :

$$0 + 2 = 2 < \infty$$

Plus court chemin

Représentation des éseaux

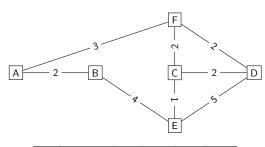
Graphe non orien Graphe pondéré

> rotocole RIP : Igorithme de ellman-Ford

rincipe

Mise en application

OSPF : algorithme



Second prédécesseur de B : E

$$\infty + 4 = \infty < \infty$$

### Plus court chemin

Représentation des réseaux

Graphe nondéré

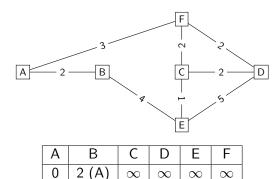
rotocole RIP : Igorithme de Jellman-Ford

rincipe

Mise en application

OSPF : algorithme

e Dijkstra



Premier prédécesseur de C : D

$$\infty + 1 = \infty \not< \infty$$

### Plus court chemin

Représentation des réseaux

Graphe non orienté

rotocole RIP : Igorithme de Jellman-Ford

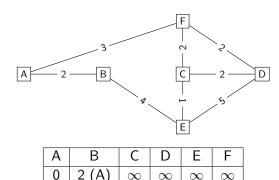
rincipe

Mise en application

OSPF: algorithme

Principe Mise en applicatio

Mise en applica Complexité



Deuxième prédécesseur de C : E

$$\infty + 1 = \infty \not< \infty$$

### Plus court chemin

Représentation des réseaux

Graphe non orient Graphe pondéré

rotocole RIP : Igorithme de Sellman-Ford

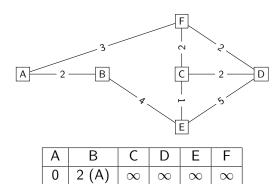
Principe

Mise en application

OSPF : algorithme de Dijkstra

Principe Mise en application

Mise en appli Complexité



Troisième prédécesseur de C : F

$$\infty + 2 = \infty \not< \infty$$

### Plus court chemin

Représentation des réseaux

Graphe non orient Graphe pondéré

rotocole RIP : Igorithme de ellman-Ford

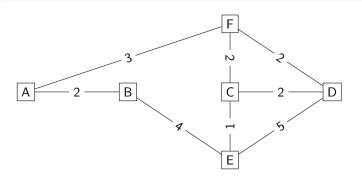
Principe

Mise en application

OSPF: algorithme

Principe

# **Activité 1 :** Continuer de dérouler la première itération de l'algorithme sur le graphe.



### Plus court chemin

Représentation des

Graphe non orienté

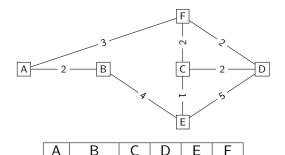
Protocole RIP : algorithme de Bellman-Ford

rincipe

Mise en application

OSPF : algorithme

Principe



 $\infty$ 

 $\infty$ 

 $\infty$ 

 $\infty$ 

2 (A)

Même constat pour D

### Plus court chemin

Représentation des

Graphe non orient

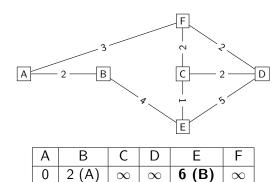
Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en application

OSPF: algorithme

Principe



Premier prédécesseur de E : B

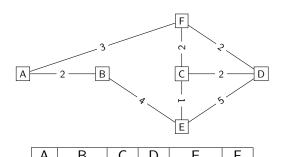
$$2 + 4 = 6 < \infty$$

 $\infty$ 

 $\infty$ 

 $\infty$ 

### Plus court chemin



Deuxième prédécesseur de E : C

2 (A)

$$\infty + 1 = \infty > 6$$

 $\infty$ 

 $\infty$ 

6 (B)

 $\infty$ 

### Plus court chemin

Représentation des réseaux

Graphe non orien Graphe pondéré

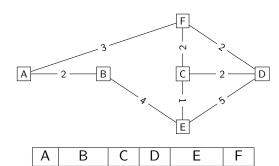
> Protocole RIP : Igorithme de Bellman-Ford

rincipe

Mise en application

OSPF: algorithme

Principe



Troisième prédécesseur de E : D

2 (A)

$$\infty + 5 = \infty > 6$$

 $\infty$ 

 $\infty$ 

6 (B)

 $\infty$ 

### Plus court chemin

Représentation des réseaux

Graphe non orien Graphe nondéré

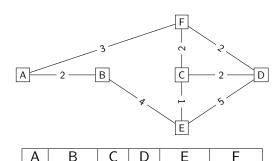
Protocole RIP Ilgorithme de Bellman-Ford

rincipe

Mise en application

OSPF : algorithme de Diikstra

Principe



Premier prédécesseur de F : A

2 (A)

$$0 + 3 = 3 < \infty$$

 $\infty$ 

 $\infty$ 

6 (B)

3 (A)

### Plus court chemin

Représentation des réseaux

Graphe non orier Graphe pondéré

> rotocole RIP : Igorithme de Jellman-Ford

rincipe

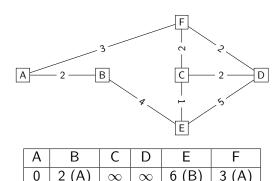
Mise en application

OSPF: algorithme de Diikstra

Principe

Mise en application

Complexité



Deuxième prédécesseur de F : C

$$\infty + 2 = \infty \nless 3$$

### Plus court chemin

Représentation des réseaux

Graphe non orier Graphe pondéré

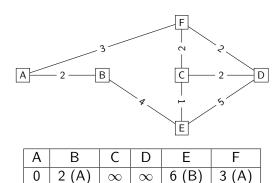
> rotocole RIP : Igorithme de ellman-Ford

rincipe

Mise en application

OSPF : algorithme

Principe



Troisième prédécesseur de F : D

$$\infty + 2 = \infty \nless 3$$

### Plus court chemin

Représentation des réseaux

Graphe non orien: Graphe pondéré

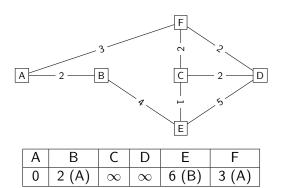
rotocole RIP : Igorithme de Jellman-Ford

rincipe

Mise en application

OSPF : algorithme

Principe



Fin de la première itération

### Plus court chemin

Représentation des éseaux

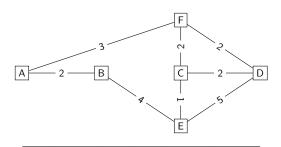
Graphe non orier Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme



 $\infty$ 

 $0 \mid 2 \text{ (A)} \mid \infty \mid \infty \mid 6 \text{ (B)} \mid 3 \text{ (A)}$  Pas de modification pour A et B

 $\infty$ 

B 2 (A)

#### Plus court chemin

Représentation des

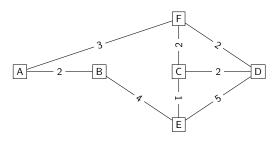
Graphe non orien Graphe nondéré

Protocole RIP : algorithme de Bellman-Ford

rincipe

Mise en application

OSPF : algorithme de Dijkstra



Α	В	C	D	Е	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	7 (E)	$\infty$	6 (B)	3 (A)

Premier prédécesseur de C : D

$$\infty + 2 = \infty \not< \infty$$

#### Plus court chemin

Représentation des éseaux

Graphe non oriente Graphe pondéré

rotocole RIP : gorithme de ellman-Ford

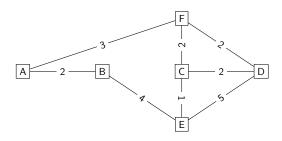
incipe

Mise en application

OSPF : algorithme

Principe

Mise en application Complexité



Α	В	С	D	Е	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	7 (E)	$\infty$	6 (B)	3 (A)

Deuxième prédécesseur de C : E

$$6 + 1 = 7 < \infty$$

#### Plus court chemin

Représentation des éseaux

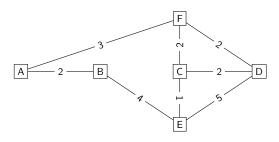
Graphe non orien Graphe pondéré

> rotocole KIP : gorithme de ellman-Ford

rincipe

Mise en application

OSPF : algorithme



Α	В	C	D	E	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	5 (F)	$\infty$	6 (B)	3 (A)

Troisième prédécesseur de C : F

$$3+2=5<7$$

#### Plus court chemin

Représentation des réseaux

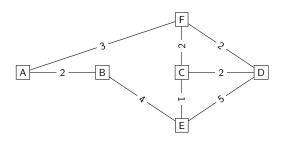
Graphe non orien: Graphe pondéré

rotocole RIP : gorithme de ellman-Ford

rincipe

Mise en application

OSPF : algorithme de Dijkstra



Α	В	С	D	Е	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Premier prédécesseur de D : C

$$5+2=7<\infty$$

#### Plus court chemin

Représentation des éseaux

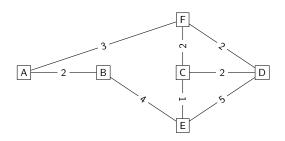
Graphe non orien: Graphe pondéré

rotocole RIP : |gorithme de |ellman-Ford

rincipe

Mise en application

OSPF : algorithme



Α	В	C	D	E	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

Second prédécesseur de D : E

$$6 + 5 = 11 \nless 7$$

#### Plus court chemin

Représentation des éseaux

Graphe non orienté Graphe pondéré

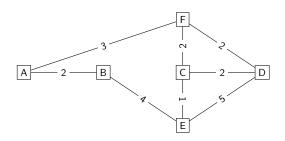
rotocole RIP : gorithme de ellman-Ford

rincipe

Mise en application

OSPF: algorithme

Principe



Α	В	C	D	E	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Troisième prédécesseur de D : F

$$3+2=5<7$$

#### Plus court chemin

Représentation des éseaux

Graphe non orien Graphe pondéré

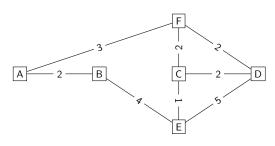
rotocole RIP : gorithme de ellman-Ford

rincipe

Mise en application

OSPF : algorithme

de Dijkstra Principe



Α	В	С	D	E	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Pas de changement pour E et F

#### Plus court chemin

Représentation des

Graphe non orien Graphe nondéré

rotocole RIP : |gorithme de ellman-Ford

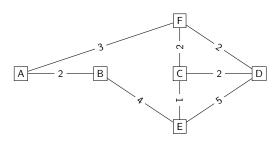
Principe

Mise en application

Complexit

OSPF : algorithme de Dijkstra

Principe Mise en annlicatio



Α	В	C	D	Е	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Fin de la deuxième itération

#### Plus court chemin

deprésentation des

Graphe non orient Graphe nondéré

rotocole RIP : Igorithme de Sellman-Ford

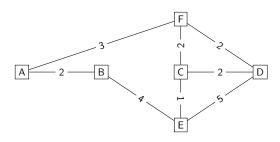
Principe

Mise en application

Complexite

OSPF : algorithme de Dijkstra

## Troisième itération



Α	В	C	D	E	F
0	2 (A)	$\infty$	$\infty$	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)
0	2 (A)	5 (F)	5 (F)	6 (B)	3 (A)

Pas de changement lors de la troisième itération

#### Plus court chemin

deprésentation des éseaux

Graphe non orient Graphe pondéré

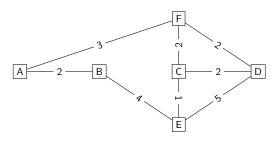
rotocole RIP : Igorithme de sellman-Ford

incipe

Mise en application

Complexité

OSPF : algorithme de Dijkstra



Α	В	C	D	Е	F
0	2 (A)	5 (F)	7 (C)	6 (B)	3 (A)

On remonte dans le tableau pour trouver le chemin vers une destination.

$$D \rightarrow C \rightarrow F \rightarrow A$$

#### Plus court chemin

Représentation des réseaux

Graphe non orient Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Mise en application

Complexité

OSPF : algorithme de Dijkstra

## Sommaire

- 1 Panrácantation des récesur
- 2. Protocole RIP: algorithme de Bellman-Ford
- 2.1 Principe
- 2.2 Mise en application
- 2.3 Complexité
- 3. OSPF: algorithme de Dijkstra

#### Plus court chemin

Représentation des

Graphe non orienté

Protocole Ri

algorithme de Bellman-Ford

Principe Mise en annlicat

Complexité

OSPF : algorithme de Diikstra

Principe

## Complexité

## La complexité dépend de :

du nombre de sommets (notée S) : on visite chaque sommet

Tant que (le nombre d'itérations) < (nombre de routeurs)

#### Plus court chemin

Représentation des éseaux

Graphe non orienté

rotocole RIP gorithme de ellman-Ford

Principe Mise en application

#### Complexité

OSPF : algorithme de Dijkstra

## La complexité dépend de :

- du nombre de sommets (notée S) : on visite chaque sommet
- Tant que (le nombre d'itérations) < (nombre de routeurs)
- du nombre d'arcs (notée A) : pour chaque sommet on regarde tous les arcs du graphe
- 1 Pour chaque arc du graphe

Représentation des éseaux

iraphe non orienté iraphe pondéré

rotocole RIP Igorithme de Jellman-Ford

Principe Mise en application

#### Complexité

OSPF : algorithme de Dijkstra

# Complexité de l'algorithme de Bellman Ford

O(S.A)

#### Plus court chemin

eprésentation des seaux

Graphe non orienté

rotocole RIP Igorithme de Bellman-Ford

Principe
Mise en application
Complexité

OSPF : algorithme de Diikstra

## Sommaire

- 1. Représentation des réseaux
- 2. Protocole RIP : algorithme de Bellman-Ford
- 3. OSPF: algorithme de Dijkstra
- 3.1 Principe
- 3.2 Mise en application
- 3.3 Complexité

#### Plus court chemin

Représentation des

Graphe non orienté

Graphe por

algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme de Dijkstra

Principe

# OSPF: algorithme de Dijkstra - principe

- Edsger Dijkstra : mathématicien néerlandais
- ► Algorithme utilisé dans GPS

#### Plus court chemin

Représentation des éseaux

Graphe non orienté

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en appl

OSPF : algorithme de Diikstra

#### Principe

# À retenir

<u>principe</u>: construire un sous-graphe en ajoutant à chaque itération un sommet de distance minimale.

#### Plus court chemin

Représentation des réseaux

Graphe non onen Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme de Dijkstra

#### Principe

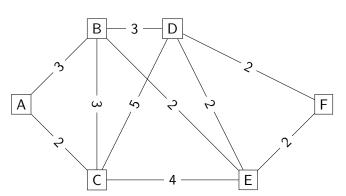


FIGURE 6 – graphe non orienté et pondéré

#### Plus court chemin

Représentation des réseaux

Graphe non orient Graphe nondéré

Protocole RIP : algorithme de Bellman-Ford

Principe

Mise en applic

OSPF : algorithme de Dijkstra

#### Principe

## Sommaire

- 1. Représentation des réseaux
- 2. Protocole RIP : algorithme de Bellman-Ford
- 3. OSPF: algorithme de Dijkstra
- 3.1 Principe
- 3.2 Mise en application
- 3.3 Complexité

#### Plus court chemin

Représentation des

Graphe non orienté

Graphe pon

algorithme de Bellman-Ford

Principe

Mise en applio

OSPF : algorithme de Diikstra

rincipe

Mise en application

# Mise en application

## Initialisation:

- Créer un tableau des distances entre A et les routeurs. initialisées à l'infini.
- Modifier la distance vers A à 0.

#### Plus court chemin

#### Initialisation:

- Créer un tableau des distances entre A et les routeurs, initialisées à l'infini.
- ► Modifier la distance vers A à 0.

## Déroulement :

- ► Tant qu'il reste des routeurs non sélectionnés
  - Parmi les routeurs non-sélectionnés, choisir celui (noté S) ayant la plus petite distance.
  - Pour chaque routeur adjacent à S (noté V) et non déjà sélectionné :
    - Si (la distance de V) > (la distance de S + poids S-V) ⇒ Mettre à jour la distance de V.

#### Représentation des éseaux

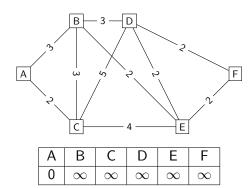
phe non orienté

rotocole RIP gorithme de ellman-Ford

Principe Mise en application

OSPF: algorithme

## Initialisation



#### Plus court chemin

Représentation des

Graphe non orien

Protocole RIP : algorithme de Bellman-Ford

Principe

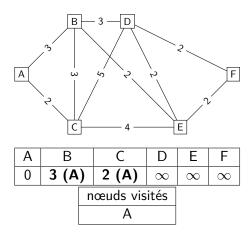
Mise en app

OSPF : algorithme

. Dijkoti d

Principe

## Sélection de A



#### Plus court chemin

Représentation des

Graphe non orien Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

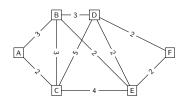
incipe

Comployitó

OSPF: algorithme

Principe

Mise en application



On sélectionne le nœud non encore visité et avec la plus petite distance : C.

Α	В	С	D	E	F
0	3 (A)	2 (A)	$\infty$	$\infty$	$\infty$
	3 (A)	2 (A)	7 (C)	6 (C)	$\infty$

nœuds visités A - C

#### Plus court chemin

Représentation des réseaux

Graphe non orient

rotocole RIP : Igorithme de

rincipe

Mise en applica

OSPF : algorithme

rincipe

Mise en application

## Observation

La route la plus courte a déjà été déterminée pour les nœuds déjà visités. Ils ne seront plus modifiés (cellule grise).

#### Plus court chemin

Représentation des éseaux

Graphe non orient

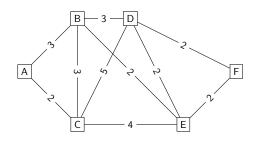
Protocole RIP Igorithme de Bellman-Ford

Mise en application

OSPF : algorithme de Dijkstra

Principe

# **Activité 2 :** Continuer de dérouler l'algorithme sur le graphe.



#### Plus court chemin

Représentation des

phe non orienté

Protocole RIP : algorithme de Bellman Ford

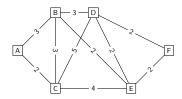
Principe

Mise en appl

de Dijkstra

Principe

Mise en application



On sélectionne le nœud non encore visité et avec la plus petite distance :  $B. \ \ \,$ 

Α	В	С	D	E	F
0	3 (A)	2 (A)	$\infty$	$\infty$	$\infty$
	3 (A)	2 (A)	7 (C)	6 (C)	$\infty$
	3 (A)		6 (B)	5 (B)	$\infty$

nœuds visités A - C - B

#### Plus court chemin

deprésentation des éseaux

Graphe non orien Graphe pondéré

rotocole RIP : Igorithme de Jellman-Ford

rincipe Ilise en applica

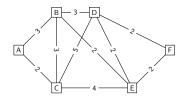
omplexité

DSPF : algorithme le Dijkstra

Principe

Mise en application

59 / 67



On sélectionne le nœud non encore visité et avec la plus petite distance : E.

Α	В	С	D	E	F
0	3 (A)	2 (A)	$\infty$	$\infty$	$\infty$
	3 (A)	2 (A)	7 (C)	6 (C)	$\infty$
	3 (A)		6 (B)	5 (B)	$\infty$
			6 (B)	5 (B)	7 (E)

nœuds visités A - C - B - E

#### Plus court chemin

eprésentation des éseaux

Graphe non orien

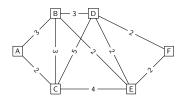
rotocole RIP :

ncipe

Mise en applica

SPF : algorithme

rincino



On sélectionne le nœud non encore visité et avec la plus petite distance :  $\mathsf{D}.$ 

Α	В	С	D	Е	F
0	3 (A)	2 (A)	$\infty$	$\infty$	$\infty$
	3 (A)	2 (A)	7 (C)	6 (C)	$\infty$
	3 (A)		6 (B)	5 (B)	$\infty$
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)

nœuds visités A - C - B - E - D Représentation des réseaux

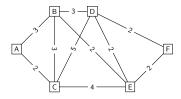
Graphe non orien Graphe pondéré

> rotocole RIP : gorithme de

lise en applicatio

CDE

JSPF : algorithme le Dijkstra



On sélectionne le nœud non encore visité et avec la plus petite distance : F.

Α	В	С	D	E	F
0	3 (A)	2 (A)	$\infty$	$\infty$	$\infty$
	3 (A)	2 (A)	7 (C)	6 (C)	$\infty$
	3 (A)		6 (B)	5 (B)	$\infty$
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)
					7 (E)

nœuds visités A - C - B - E - D - F

#### Plus court chemin

Représentation des éseaux

Graphe non orien Graphe pondéré

> rotocole RIP gorithme de ellman-Ford

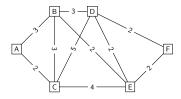
rincine

lise en applicatio

SPF : algorithme e Diikstra

rincipe

Mise en application



On sélectionne le nœud non encore visité et avec la plus petite distance : F.

Α	В	С	D	Е	F
0	3 (A)	2 (A)	$\infty$	$\infty$	$\infty$
	3 (A)	2 (A)	7 (C)	6 (C)	$\infty$
	3 (A)		6 (B)	5 (B)	$\infty$
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)
					7 (E)

nœuds visités
A - C - B - E - D - F

Plus court chemin

Représentation des réseaux

Graphe non orien Graphe nondéré

rotocole RIP : gorithme de ellman-Ford

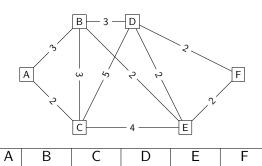
rincipe

Mise en application

SPF: algorithme

rincipe

Mise en application



3 (A) 2 (A) 6 (B) 5 (B)

On remonte dans le tableau pour trouver le chemin vers une destination.

$$F \rightarrow E \rightarrow B \rightarrow A$$

#### Plus court chemin

## Sommaire

- 1. Représentation des réseaux
- 2. Protocole RIP : algorithme de Bellman-Forc
- 3. OSPF: algorithme de Dijkstra
- 3.1 Principe
- 3.2 Mise en application
- 3.3 Complexité

#### Plus court chemin

Représentation des

Graphe non orienté

Graphe pon

algorithme de Bellman-Ford

Principe

Mise en applica

OSPF : algorithme

Principe

Mise en application

# Complexité

La complexité dépend du nombre de sommets S et du nombre d'arcs A.

#### Plus court chemin

Représentation des

Graphe non orienté

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en appli

OSPF : algorithme de Diikstra

Principe
Mise en application
Complexité

- La complexité dépend du nombre de sommets S et du nombre d'arcs A.
- Le point clé de l'algorithme tient dans la recherche de la distance minimale.
  - Parmi les routeurs non—sélectionnés, choisir le routeur (noté S) ayant la plus petite distance.

Représentation des éseaux

Graphe non onent

Protocole RIP

Principe

Mise en application

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité

# Complexité de l'algorithme de Dijkstra

$$O((A + S) \times \log S)$$

#### Plus court chemin

Représentation des

Graphe non orienté

Protocole RIP algorithme de Bellman-Ford

Principe Mise en applicatio

OSPE : algorithme

Principe
Mise en application