Plus court chemin

Graphe orienté

Protocole RIP algorithme de Bellman-Ford

Principe
Mise en applicatio
Complexité

OSPF : algorithme de Dijkstra

Principe Mise en applicatio

Plus court chemin

Christophe Viroulaud

Terminale NSI

Plus court chemin

Problématique

Retour des graphes Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

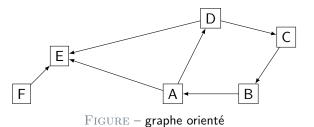
Principe Mise en application

Comment fonctionnent les algorithmes de plus court chemin?

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité

Graphe orienté / non orienté



Plus court chemin

Problématiqu

Retour des graphes

Graphe orienté

Protocole RIP algorithme de Bellman-Ford

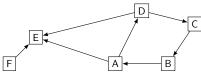
rincipe Mise en application

Complexité

OSPF : algorithm

de Dijkstra Principe

Mise en application



 $\mathrm{FIGURE}-\text{graphe orient\'e}$

À retenir

Pour chaque nœud on peut définir ses **prédécesseurs** et ses **successeurs**.

Plus court chemin

Problématique

Retour des graph

Graphe orienté

Protocole RIP

Principe Mise en application Complevité

OSPF : algorithme

Principe
Mise en application
Complevité

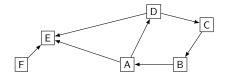
Plus court chemin

Graphe orienté

Activité 1:

- 1. Établir la matrice d'adjacence du graphe figure 2.
- 2. Établir le dictionnaire des successeurs du graphe figure 2.
- 3. Déterminer un cycle dans le graphe.

Correction



Code 1 – Matrice d'adjacence des successeurs

Plus court chemin

Problématique

Retour des graphe

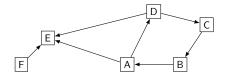
Graphe orienté

Protocole RIP Igorithme de

Principe Mise en application

OSPF : algorithme

Correction



Code 2 – Matrice d'adjacence des **prédécesseurs**

Plus court chemin

Problématique

Retour des graphe

Graphe orienté

rotocole RI

incipe

Complexité

OSPE : algorithm

Principe
Mise en application

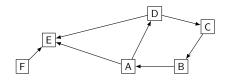
Mise en application

```
dico = \{"A": \{"D", "E"\},
           "B": {"A"},
           "C": {"B"},
3
           "D": {"C", "E"},
4
           "E": set(),
           "F": {"E"} }
6
```

Code 3 – Dictionnaire d'adjacence des successeurs

Plus court chemin

Graphe orienté



Code 4 – Dictionnaire d'adjacence des **prédécesseurs**

rrobiematique

Retour des graphes

Graphe orienté

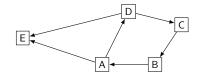
Protocole RI

ellman-Ford

Complexité

OSPF : algorithme de Dijkstra

Correction



Un cycle

$$\mathsf{A}\to\mathsf{D}\to\mathsf{C}\to\mathsf{B}\to\mathsf{A}$$

Plus court chemin

Problématiqu

Retour des graphes

Graphe orienté

Protocole RIP algorithme de

Principe Mise en application

omplexité

OSPF : algorithme de Dijkstra

Plus court chemin

Problématique

Retour des graphes

Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Principe Mise en applicatio

OSPF : algorithmede Dijkstra

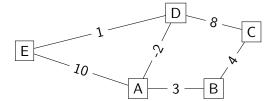


FIGURE – graphe non orienté pondéré

Plus court chemin

Problématique

Graphe orienté

Graphe pondéré

algorithme de Bellman-Ford

Mise en application Complexité

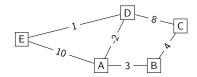
OSPF : algorithme de Dijkstra

Principe Mise en application

Activité 2:

- 1. Établir la matrice d'adjacence du graphe figure 3.
- 2. Établir le dictionnaire d'adjacence du graphe figure 3.

Correction



Code 5 – Matrice d'adjacence

Plus court chemin

Problématique

Retour des graphes Graphe orienté

Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

> ise en application omplexité

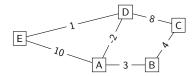
OSPF : algorithme de Dijkstra

Principe

Mise en application

Complexité

Correction



```
1 dico = {"A": {"B": 3, "D": -2, "E": 10},

2 "B": {"A": 3, "C": 4},

3 "C": {"B": 4, "D": 8},

4 "D": {"A": -2, "C": 8, "E": 1},

5 "E": {"D": 1, "E": 10}}
```

Code 6 - Dictionnaire d'adjacence

Plus court chemin

Problématique

Retour des graphes

Graphe pondéré

algorithme de Bellman-Ford

> ise en application omplexité

OSPF : algorithme de Dijkstra

Algorithme de Bellman-Ford : Fin des années 50

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

roblématique?

Retour des gr Graphe orienté

Protocole RIP

Principe

Mise en application

OSPF : algorithme de Diikstra

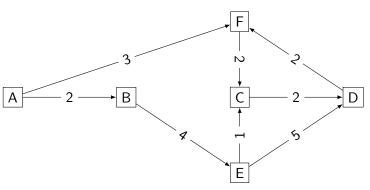


FIGURE – graphe orienté pondéré

En pratique nous allons utiliser une approche itérative *bottom-up*.

Problématiqu

Retour des graphes Graphe orienté Graphe pondéré

Protocole RIP : algorithme de Rollman Ford

Principe

Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité

Pour chaque routeur

On obtient un tableau contenant la distance minimale entre le routeur de départ et chaque autre routeur.

Plus court chemin

roblématique

Retour des graphe Graphe orienté

orapiic policic

algorithme de Bellman-Ford

Principe

Mise en application Complexité

OSPF : algorithme de Diikstra

Mise en application

Code 7 – Algorithme de Bellman Ford

Si (la distance du routeur) > (la distance de son prédé

La distance du routeur est remplacée par cette

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

cesseur + poids de l'arc entre les deux routeurs)

Créer un tableau des distances entre A et les routeurs (A inclus Dans le tableau modifier la distance vers A à 0.

), initialisées à l'infini.

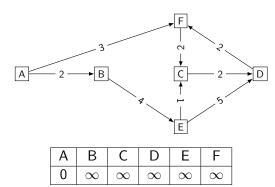
Pour chaque arc du graphe

nouvelle valeur

3

8

Initialisation



Plus court chemin

Problématique

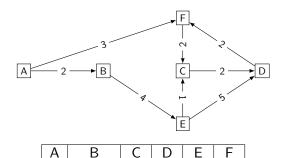
Retour des graphes Graphe orienté

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme



 ∞

 ∞

 ∞

 ∞

2 (A)

Plus court chemin

Problématique

Retour des graphes Graphe orienté

Protocole RIP algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme de Diikstra

Activité 3 : Continuer de dérouler l'algorithme sur le graphe figure 4.

Plus court chemin

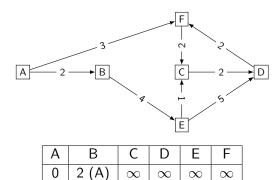
Problématique

Retour des graphes
Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithm



Premier prédécesseur de C : E

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

Plus court chemin

Problématique

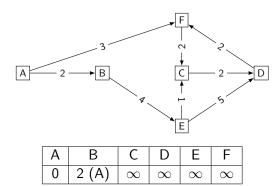
Retour des graphes Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Principe Mise en application

Complexité

OSPF : algorithme de Dijkstra



Second prédécesseur de C : F

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

Plus court chemin

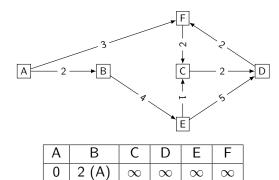
Problématique

Retour des graphes
Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application Complexité

OSPF : algorithme de Dijkstra



Même constat pour D

Plus court chemin

Problématique

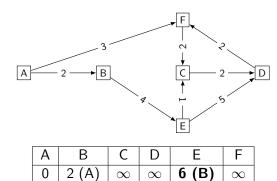
Retour des graphes Graphe orienté

Protocole RIP algorithme de Bellman-Ford

Principe Mise en application

Mise en application

OSPF : algorithme de Dijkstra



Prédécesseur de E : B

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

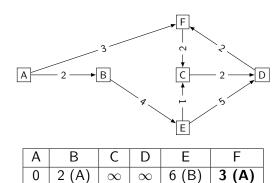
 ∞

 ∞

 ∞

Plus court chemin

Mise en application



Premier prédécesseur de F: A

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

Plus court chemin

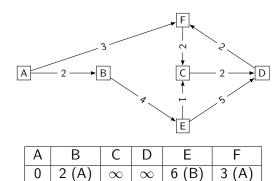
Problématique

Retour des graphes
Graphe orienté
Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithme de Dijkstra



Second prédécesseur de F : D

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

Plus court chemin

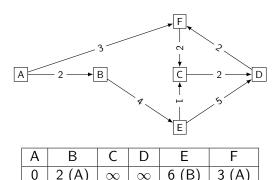
Problématique

Retour des graphes
Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application Complexité

OSPF : algorithme de Dijkstra



Fin de la première itération

Plus court chemin

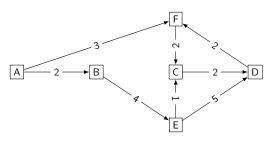
Problématique

Retour des graphe Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithme de Dijkstra



Α	В	С	D	Е	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	∞	∞	6 (B)	3 (A)

Pas de modification pour A et B

Plus court chemin

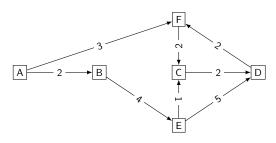
Problématique

Retour des graphes Graphe orienté

Protocole RIP : Ilgorithme de Bellman-Ford

Mise en application

OSPF : algorithme



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

Premier prédécesseur de C : E

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

Plus court chemin

Problématique

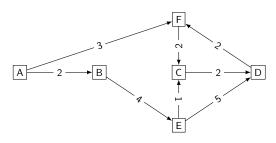
Retour des graphes Graphe orienté

rotocole RIP : Igorithme de Sellman-Ford

Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité



Α	В	С	D	E	F
0	2 (A)	∞	∞	6 (B)	3 (A)
0	2 (A)	5 (F)	∞	6 (B)	3 (A)

Second prédécesseur de C : F

$$3+2=5<7$$

Plus court chemin

Problématique

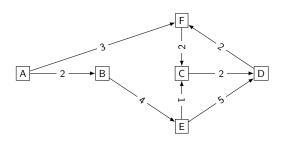
Retour des graphes Graphe orienté

Protocole RIP : Igorithme de Bellman-Ford

Mise en application
Complexité

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité



	Α	В	C	D	Е	F
Ī	0	2 (A)	∞	∞	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

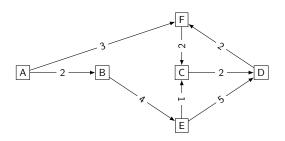
Problématique

Retour des graphes Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithme



Α	В	C	D	E	F
0	2 (A)	∞	∞	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

Problématique

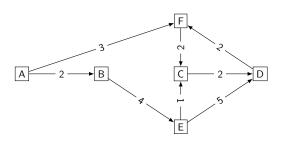
Retour des graphes Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application
Complexité

OSPF : algorithme

Principe
Mise en application
Complexité



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

Pas de changement pour E et F

Plus court chemin

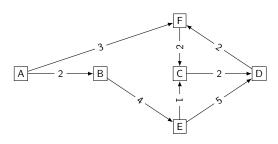
Problématiqu

Retour des graphes
Graphe orienté
Graphe pondéré

lgorithme de Bellman-Ford

Mise en application

OSPF : algorithme



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

Fin de la deuxième itération

Plus court chemin

Problématique

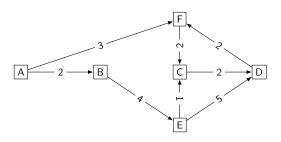
Retour des graphes
Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithme

Troisième itération



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

Pas de changement lors de la troisième itération

Plus court chemin

Problématique

Retour des graphes Graphe orienté

rotocole RIP : Igorithme de Sellman-Ford

Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité

 du nombre de sommets (notée S) : on visite chaque sommet (ligne 4);

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

Problématique

Retour des graphes Graphe orienté Graphe pondéré

algorithme de Bellman-Ford Principe

Mise en application Complexité

OSPF : algorithme de Dijkstra

1 | 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 (ligne 5).
 - 1 Pour chaque arc du graphe

Problématique

Retour des graphes Graphe orienté Graphe pondéré

Bellman-Ford
Principe
Mise en application

Complexité OSPF: algorithme

Complexité de l'algorithme de Bellman Ford

O(S.A)

Plus court chemin

Problématique

Retour des graphes Graphe orienté

Graphe pondéré

Bellman-Ford

Mise en application

Complexité

OSPF: algorithme

Edsger Dijkstra

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

Plus court chemin

Problématique

Retour des graphes Graphe orienté

algorithme de Bellman-Ford

Principe
Mise en application
Complexité

OSPF : algorithme de Diikstra

Principe

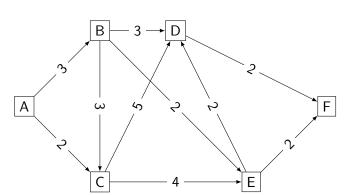


FIGURE – graphe orienté et pondéré

Plus court chemin

Problématique

Retour des graphes
Graphe orienté
Graphe pondéré

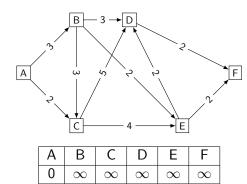
algorithme de Bellman-Ford

Vise en application Complexité

OSPF : algorithme de Dijkstra

- Créer un tableau des distances entre A et les routeurs (A inclus), initialisées à l'infini.
- Dans le tableau modifier la distance vers A à 0.
 - Tant qu'il reste des routeurs non sélectionnés
 - Parmi les routeurs non-sélectionnés, choisir le routeur (noté S) ayant la plus petite distance.
 - Pour chaque routeur adjacent à S (noté V) et non déjà sé lectionné:
 - Si (la distance de V) > (la distance de S + poids S-V) La distance de V est remplacée par cette nouvelle valeur

Initialisation



Plus court chemin

Problématiqu

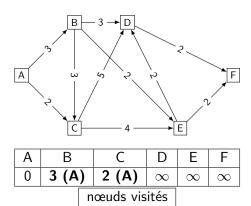
Retour des graphes Graphe orienté

algorithme de Bellman-Ford

Principe Mise en application Complexité

OSPF : algorithme de Dijkstra

Sélection de A



Α

Plus court chemin

Problématiqu

Retour des graphes
Graphe orienté

Bellman-Ford Principe

lise en application omplexité

OSPF : algorithme de Dijkstra

Mise en application

Activité 4 : Continuer de dérouler l'algorithme sur le graphe figure 5.

Plus court chemin

Problématique

Retour des graphes

Graphe orienté

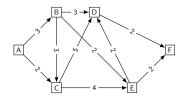
Graphe pondéré

algorithme de Bellman-Ford

Principe Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe

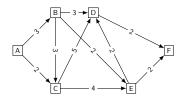


On sélectionne le nœud non encore visité et avec la plus petite distance : C.

Α	В	С	D	Е	F
0	3 (A)	2 (A)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞

nœuds visités A - C

Plus court chemin



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

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

nœuds visités A - C - B Plus court chemin

Problématique

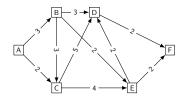
Retour des graphes Graphe orienté

> Igorithme de sellman-Ford Principe

Complexité DSPF : algorithme

Principe
Mise en application

Mise en application Complexité



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)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞
			6 (B)	5 (B)	7 (E)

nœuds visités A - C - B - E Plus court chemin

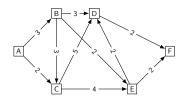
Problématique

Retour des graphes
Graphe orienté
Graphe pondéré

algorithme de Bellman-Ford Principe

> ise en application omplexité

OSPF : algorithme de Dijkstra



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

В	C	D	Е	F
3 (A)	2 (A)	∞	∞	∞
3 (A)	2 (A)	7 (C)	6 (C)	∞
3 (A)		6 (B)	5 (B)	∞
		6 (B)	5 (B)	7 (E)
		6 (B)		7 (E)
	3 (A) 3 (A)	3 (A) 2 (A) 3 (A) 2 (A)	3 (A) 2 (A) ∞ 3 (A) 2 (A) 7 (C) 3 (A) 6 (B) 6 (B)	3 (A) 2 (A) ∞ ∞ 3 (A) 2 (A) 7 (C) 6 (C) 3 (A) 6 (B) 5 (B) 6 (B) 5 (B) 6 (B)

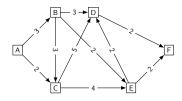
nœuds visités A - C - B - E - D Problématique

Retour des graphe Graphe orienté Graphe pondéré

Igorithme de Jellman-Ford Principe

omplexité

JSPF : algorithme de Dijkstra Principe



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)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)
					7 (E)

nœuds visités A - C - B - E - D - F Plus court chemin

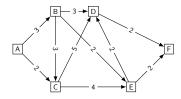
Problématique

Retour des graphe Graphe orienté Graphe pondéré

gorithme de ellman-Ford

ise en application omplexité

OSPF : algorithme de Dijkstra



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)	∞	∞	∞
	3 (A)	2 (A)	7 (C)	6 (C)	∞
	3 (A)		6 (B)	5 (B)	∞
			6 (B)	5 (B)	7 (E)
			6 (B)		7 (E)
					7 (E)

nœuds visités
A - C - B - E - D - F

Plus court chemin

Problématiqu

Retour des graphe
Graphe orienté
Graphe pondéré

gorithme de ellman-Ford

ise en application omplexité

OSPF : algorithme e Dijkstra

Plus court chemin

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

Problématique

Graphe orienté

Graphe pondéré

Protocole RIP : algorithme de Bellman-Ford

Principe Vlise en application Complexité

OSPF : algorithme de Dijkstra

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.

Problématique

Graphe orienté
Graphe pondéré

algorithme de Bellman-Ford Principe Mise en application

OSPF : algorithme de Diikstra

Principe
Mise en application
Complexité

Complexité de l'algorithme de Dijkstra

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

Plus court chemin

Problématique

Retour des graphes Graphe orienté

Protocole RIP : algorithme de

Principe Mise en application

OSPF: algorithme

Principe Mise en application Complexité