Plus court chemin

Plus court chemin

Christophe Viroulaud

Terminale NSI

Problématique

Retour des graphes

Graphe pondér

Protocole RIP algorithme de Bellman-Ford

Principe Mise en application Complexité

OSPF : algorithme de Dijkstra

Plus court chemin

Problématique

Retour des graphes Graphe orienté

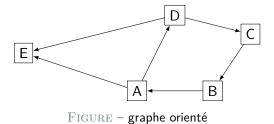
Protocole RIP : algorithme de Bellman-Ford

Mise en application

Comment fonctionnent les algorithmes de plus court chemin?

OSPF : algorithme de Dijkstra

Graphe orienté / non orienté



Plus court chemin

Problématique

Retour des graphes

Graphe orienté

Protocole RIF

Bellman-Ford Principe

Mise en application Complexité

OSPF : algorithme de Dijkstra

E A B

 $\label{eq:figure} Figure - \text{graphe orient\'e}$

À retenir

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

Plus court chemin

Problématique

Retour des graphe

Graphe orienté

Protocole RIP algorithme de

Principe Mise en application

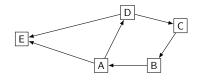
OSPF : algorithme de Dijkstra

Plus court chemin

Graphe orienté

Activité 1:

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



Code 1 – Matrice d'adjacence

Plus court chemin

Problématique |

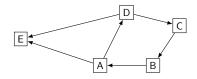
Retour des graphe

Graphe orienté

Protocole RIP algorithme de

rincipe lise en application

OSPF : algorithme



Code 2 – Dictionnaire d'adjacence

Plus court chemin

Problématique

Retour des graphe

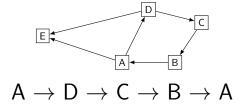
Graphe orienté

Protocole RIP algorithme de

Principe Mise en application

OSPF : algorithme

de Dijkstra Principe



Plus court chemin

Problématiqu

Retour des graphes

Graphe orienté

Protocole RIP algorithme de Bellman-Ford

Principe Mise en application

OSPF : algorithme

de Dijkstra Principe

Principe
Mise en application
Complevité

Plus court chemin

Problématique

Retour des graphes

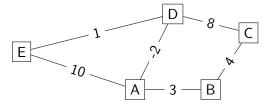
Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

Principe Mise en applicatio

OSPF : algorithme

Principe
Mise en application



Plus court chemin

Problematique

Graphe orienté

Graphe pondéré

algorithme de Bellman-Ford

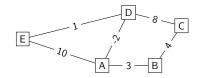
Mise en application Complexité

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.



Code 3 – Matrice d'adjacence

Plus court chemin

Problématique

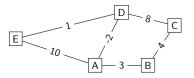
Retour des graphes

Graphe pondéré

Protocole RIP Ilgorithme de Bellman-Ford

lise en application omplexité

DSPF : algorithme de Dijkstra



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

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

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

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

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

Code 4 – Dictionnaire d'adjacence

Problématique

Ketour des graphes

Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

se en application

OSPF : algorithme de Dijkstra

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.

Problématique

Retour des graphe Graphe orienté

Graphe ponder

algorithme de Bellman-Ford

Principe

Mise en application

OSPF : algorithme de Diikstra

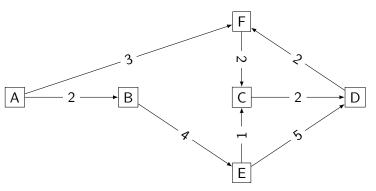


FIGURE – graphe non orienté pondéré

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

Problématique

Retour des graphes Graphe orienté Graphe pondéré

Protocole RIP : algorithme de

Principe

Vise 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 graphes Graphe orienté

Graphe pondéré

algorithme de Bellman-Ford

Principe

Mise en application Complexité

OSPF : algorithme de Dijkstra

algorithme de Bellman-Ford

Mise en application

OSPF : algorithm

Principe Mise en application

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.

Jans le tableau modifier la distance vers A a 0

Tant que (le nombre d'itérations) < (nombre de routeurs) Pour chaque arc du graphe

Si (la distance du routeur) > (la distance de son prédé cesseur + poids de l'arc entre les deux routeurs)

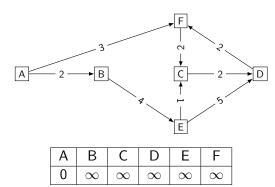
Code 5 – Algorithme de Bellman Ford

La distance du routeur est remplacée par cette nouvelle valeur

8

3

Initialisation



Plus court chemin

Problématiqu

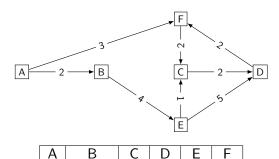
Retour des graphes Graphe orienté

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

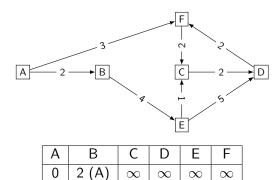
Principe

Mise en application

OSPF : algorithme de Dijkstra

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

Plus court chemin



Premier prédécesseur de C : E

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

Plus court chemin

Problématique

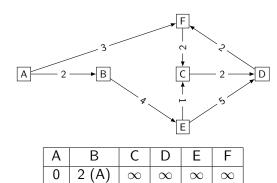
Retour des graphes Graphe orienté

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

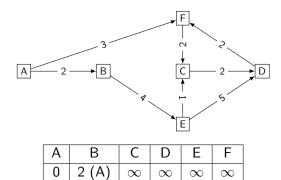
Problématique

Retour des graphes
Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

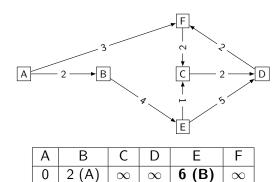
Mise en application
Complexité

OSPF : algorithme



Même constat pour D

Plus court chemin



Prédécesseur de E : B

2 (A)

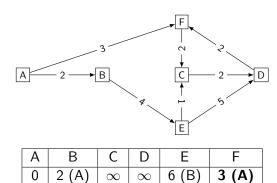
$$2+4=6<\infty$$

 ∞

 ∞

 ∞

Plus court chemin



Premier prédécesseur de F : A

2 (A)

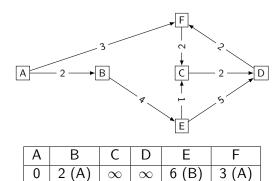
$$0 + 3 = 3 < \infty$$

 ∞

 ∞

6 (B)

Plus court chemin



Second prédécesseur de F : D

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

Plus court chemin

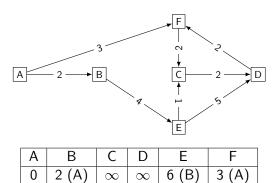
Problématique

Graphe orienté

Protocole RIP : Ilgorithme de Bellman-Ford

Mise en application Complexité

OSPF : algorithme de Dijkstra



Fin de la première itération

Plus court chemin

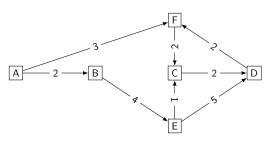
Problématique

Retour des graphes 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

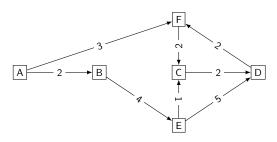
Problématique

Retour des graphes Graphe orienté

Protocole RIP : Ilgorithme de Bellman-Ford

Mise en application

OSPF : algorithme



Α	В	С	D	E	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

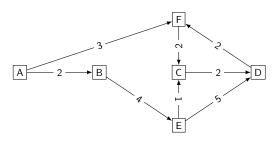
Problématique

Retour des graphes Graphe orienté

> rotocole RIP : Igorithme de Sellman-Ford

Mise en application

OSPF : algorithme de Dijkstra



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

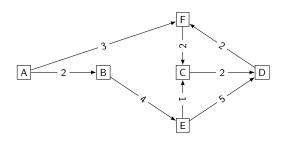
Retour des graphes Graphe orienté

Protocole RIP : Igorithme de Bellman-Ford

Mise en application

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité



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

Premier prédécesseur de D : C

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

Problématique

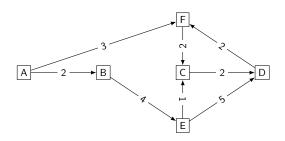
Retour des graphes Graphe orienté

rotocole RIP : Igorithme de Sellman-Ford

Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité



Α	В	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

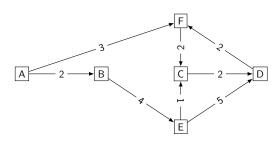
Problématique

Retour des graphes Graphe orienté

Protocole RIP : algorithme de Bellman-Ford

Mise en application

OSPF : algorithme de Dijkstra



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

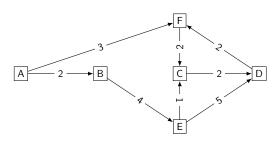
Problématiqu

Retour des graphes
Graphe orienté
Graphe pondéré

Protocole RIP : algorithme 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

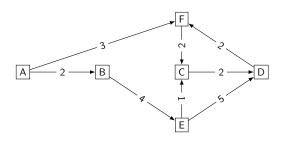
Retour des graphes Graphe orienté Graphe pondéré

Protocole RIP : Ilgorithme 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

Protocole RIP : Igorithme de Bellman-Ford

Mise en application Complexité

OSPF : algorithme de Dijkstra

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

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

'roblèmatique

Retour des graphes
Graphe orienté

algorithme de Bellman-Ford

Mise en application

Complexité

OSPF : algorithme de Dijkstra

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

```
Tant que (le nombre d'itérations)
< (nombre de routeurs)
2
```

du nombre d'arcs (notée A) : pour chaque sommet on regarde tous les arcs du graphe (ligne 5).

```
Pour chaque arc du graphe
```

Problématique

Retour des graphes
Graphe orienté
Graphe pondéré

dellman-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

algorithme de Bellman-Ford

Principe Mise en application

Complexité

OSPF: algorithme

Principe Mise en application

Edsger Dijkstra

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

Plus court chemin

Principe

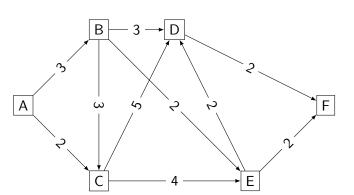


FIGURE – graphe orienté et pondéré

Plus court chemin

Problématique

Retour des graphes
Graphe orienté

algorithme de Bellman-Ford

Vise en application Complexité

OSPF : algorithme de Dijkstra

Principe Mise en application

Mise en application

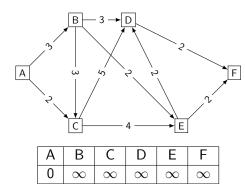
Code 6 – Algorithme de Dijkstra

valeur

9

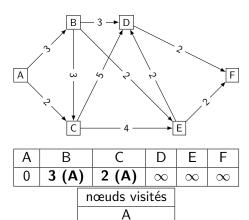
La distance de V est remplacée par cette nouvelle

Initialisation



Plus court chemin

Sélection de A



Plus court chemin

Problématiqu

Retour des graphes Graphe orienté

algorithme de Bellman-Ford Principe

Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe

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

Plus court chemin

Problématique

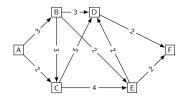
Retour des graphes
Graphe orienté
Graphe pondéré

algorithme de Bellman-Ford

Principe Mise en application Complexité

OSPF : algorithme de Dijkstra

Principe

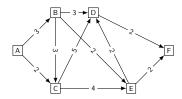


On sélectionne le nœud non encore visité et avec la plus petite distance : C.

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	E	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

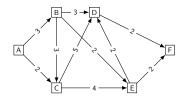
Problématique

Retour des graphe Graphe orienté

gorithme de ellman-Ford Principe

Complexité

de Dijkstra Principe



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

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)

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

Problématique

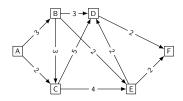
Retour des graphes Graphe orienté

algorithme de Bellman-Ford Principe

Complexité

Principe

Mise en application Complexité



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

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)

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

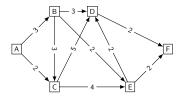
Problématique

Retour des graphe Graphe orienté Graphe pondéré

gorithme de ellman-Ford Principe

omplexité

OSPF : algorithme de Dijkstra Principa



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

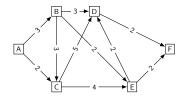
Problématique

Retour des graphe Graphe orienté Graphe pondéré

lgorithme de sellman-Ford Principe

lise en application omplexité

OSPF : algorithme de Dijk<mark>s</mark>tra



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

Igorithme de Bellman-Ford Principe

lise en application omplexité

OSPF: algorithme le Dijkstra

Plus court chemin

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

Problématique

Retour des graphes
Graphe orienté
Graphe pondéré

Protocole RIP algorithme de Bellman-Ford

^orincipe Vlise en application Complexité

OSPF : algorithme de Dijkstra

Principe
Mise en application
Complexité

nombre d'arcs A.

Le point clé de l'algorithme tient dans la recherche de

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.

roblématique

Retour des graphes Graphe orienté Graphe pondéré

lgorithme de ellman-Ford Principe Mise en application

OSPF : algorithme

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

Graphe pondé

Protocole RIP algorithme de Bellman-Ford

Principe Mise en application

OSPF: algorithme

Principe
Mise en application
Complexité