

# Graphes

## 7. Recherche de plus courts chemins

Solen Quiniou

`solen.quiniou@univ-nantes.fr`

IUT de Nantes

Année 2023-2024 – BUT 1 (Semestre 2)

[Mise à jour du 24 janvier 2024]



IUT Nantes  
Pôle Sciences et technologie

**Nantes Université**

# Plan du cours

- 1 Introduction
- 2 Distance et plus courts chemins
- 3 Algorithme de Dijkstra

# Recherche de plus courts chemins

## Exemples d'applications

- Recherche de l'itinéraire le plus rapide en voiture, entre deux villes
- Routage dans des réseaux de communication
- Problèmes d'ordonnancement. . .

## Algorithmes et cas considérés

# Recherche de plus courts chemins

## Exemples d'applications

- Recherche de l'itinéraire le plus rapide en voiture, entre deux villes
- Routage dans des réseaux de communication
- Problèmes d'ordonnancement. . .

## Algorithmes et cas considérés

- Algorithmes étudiés pour résoudre le problème suivant :

Étant donné un sommet  $x$ , on veut déterminer, pour chaque sommet  $y$ , la distance et un plus court chemin de  $x$  à  $y$

# Recherche de plus courts chemins

## Exemples d'applications

- Recherche de l'itinéraire le plus rapide en voiture, entre deux villes
- Routage dans des réseaux de communication
- Problèmes d'ordonnancement. . .

## Algorithmes et cas considérés

- Algorithmes étudiés pour résoudre le problème suivant :

Étant donné un sommet  $x$ , on veut déterminer, pour chaque sommet  $y$ , la distance et un plus court chemin de  $x$  à  $y$

- Étant donné deux sommets  $x$  et  $y$ , il existe plusieurs cas :
  - 1 Il n'existe pas de chemins de  $x$  à  $y$
  - 2 Il existe un ou plusieurs plus courts chemins de  $x$  à  $y$
  - 3 Il existe des chemins de  $x$  à  $y$  mais pas de plus courts

# Recherche de plus courts chemins

## Exemples d'applications

- Recherche de l'itinéraire le plus rapide en voiture, entre deux villes
- Routage dans des réseaux de communication
- Problèmes d'ordonnancement. . .

## Algorithmes et cas considérés

- Algorithmes étudiés pour résoudre le problème suivant :

Étant donné un sommet  $x$ , on veut déterminer, pour chaque sommet  $y$ , la distance et un plus court chemin de  $x$  à  $y$

- Étant donné deux sommets  $x$  et  $y$ , il existe plusieurs cas :
  - 1 Il n'existe pas de chemins de  $x$  à  $y$
  - 2 Il existe un ou plusieurs plus courts chemins de  $x$  à  $y$
  - 3 Il existe des chemins de  $x$  à  $y$  mais pas de plus courts

→ Problématiques similaires pour les graphes non orientés

# Plan du cours

1 Introduction

2 Distance et plus courts chemins

3 Algorithme de Dijkstra

# Distance et plus court chemin

## Définitions : distance et plus court chemin

Soit  $G = (S, A, v)$  un graphe valué et soient  $x$  et  $y$  deux sommets de  $G$

- **Distance de  $x$  à  $y$** , notée  $d(x, y)$  : minimum des valuations des chemins allant de  $x$  à  $y$
- **Plus court chemin de  $x$  à  $y$**  : tout chemin dont la valuation est égale à  $d(x, y)$

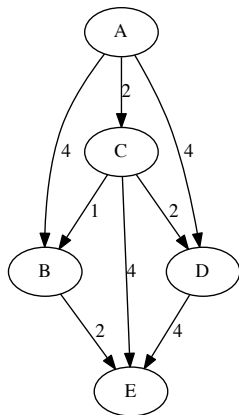
→ Définitions similaires pour les graphes non orientés

## Remarque

**Distance en nombre d'arcs** : cas particulier du calcul de distance, qui correspond au cas où tous les arcs sont de valuation 1

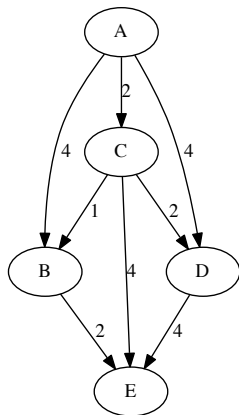


# Exemples de plus courts chemins



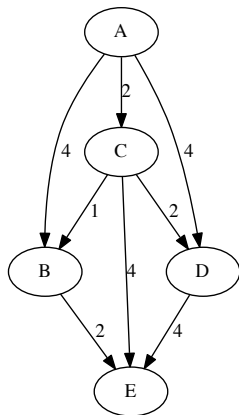
- Distance de  $A$  à  $E$  :  $d(A, E) = 5$   
→ Plus court(s) chemin(s) :  $[A, C, B, E]$

# Exemples de plus courts chemins



- Distance de  $A$  à  $E$  :  $d(A, E) = 5$   
→ Plus court(s) chemin(s) :  $[A, C, B, E]$
- Distance de  $A$  à  $D$  :  $d(A, D) = 4$   
→ Plus court(s) chemin(s) :  $[A, D]$  et  $[A, C, D]$

# Exemples de plus courts chemins



- Distance de  $A$  à  $E$  :  $d(A, E) = 5$   
→ Plus court(s) chemin(s) :  $[A, C, B, E]$
- Distance de  $A$  à  $D$  :  $d(A, D) = 4$   
→ Plus court(s) chemin(s) :  $[A, D]$  et  $[A, C, D]$
- Distance de  $E$  à  $A$  : non définie  
→ Plus court(s) chemin(s) : aucun

# Circuit absorbant

## Définition : circuit absorbant

**Circuit absorbant** : circuit de valuation négative

→ Si un graphe possède un circuit absorbant alors il n'existe pas de plus courts chemins entre certains de ses sommets

## Théorème

Soit  $G$  un graphe orienté valué sans circuit absorbant et  $x$  et  $y$  deux sommets de  $G$

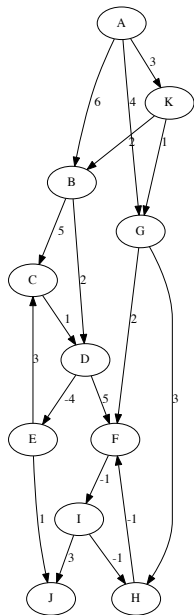
S'il existe un chemin allant de  $x$  à  $y$  alors la distance  $d(x, y)$  est bien définie et il existe au moins un plus court chemin de  $x$  à  $y$

→ Définition et théorème similaires pour les graphes non orientés

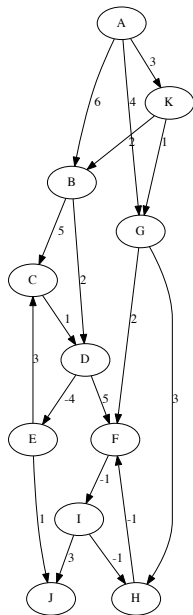
**Graphes considérés dans la suite : sans circuits (ou cycles) absorbants**

# Exemple

- De  $A$  à  $B$  : il existe **un unique plus court chemin**,  $[A, K, B]$

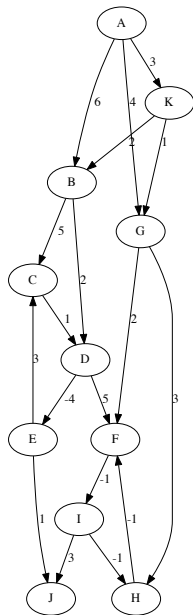


# Exemple



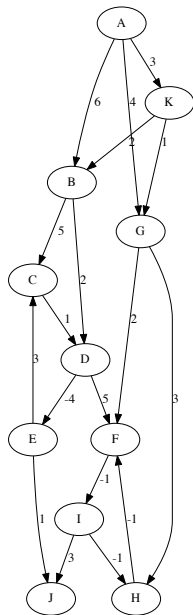
- De  $A$  à  $B$  : il existe **un unique plus court chemin**,  $[A, K, B]$
- De  $A$  à  $G$  : il existe **2 plus courts chemins**,  $[A, K, G]$  et  $[A, G]$

# Exemple



- De  $A$  à  $B$  : il existe **un unique plus court chemin**,  $[A, K, B]$
- De  $A$  à  $G$  : il existe **2 plus courts chemins**,  $[A, K, G]$  et  $[A, G]$
- De  $E$  à  $A$  : il n'existe pas de chemin donc **aucun plus court chemin**

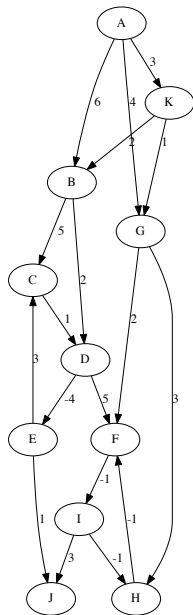
# Exemple



- De  $A$  à  $B$  : il existe **un unique plus court chemin**,  $[A, K, B]$
- De  $A$  à  $G$  : il existe **2 plus courts chemins**,  $[A, K, G]$  et  $[A, G]$
- De  $E$  à  $A$  : il n'existe pas de chemin donc **aucun plus court chemin**
- De  $A$  à  $E$  : il existe une **infinité de plus courts chemins**, tels que  $[A, K, B, D, E]$ ,  $[A, K, B, D, E, C, D, E]$ , à-cause du circuit  $[C, D, E, C]$  de valuation nulle



# Exemple



- De  $A$  à  $B$  : il existe **un unique plus court chemin**,  $[A, K, B]$
- De  $A$  à  $G$  : il existe **2 plus courts chemins**,  $[A, K, G]$  et  $[A, G]$
- De  $E$  à  $A$  : il n'existe pas de chemin donc **aucun plus court chemin**
- De  $A$  à  $E$  : il existe une **infinité de plus courts chemins**, tels que  $[A, K, B, D, E]$ ,  $[A, K, B, D, E, C, D, E]$ , à-cause du circuit  $[C, D, E, C]$  de valuation nulle
- De  $A$  à  $J$  : il existe des chemins mais **aucun plus court chemin**, à-cause du **circuit absorbant**  $[F, I, H, F]$  qui réduit la valuation du chemin de  $A$  à  $J$  à chaque passage dans le circuit

# Plan du cours

- 1 Introduction
- 2 Distance et plus courts chemins
- 3 Algorithme de Dijkstra**

# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les plus connus et efficaces pour la recherche de distance et de plus courts chemins

## Principe de l'algorithme de Dijkstra

# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les plus connus et efficaces pour la recherche de distance et de plus courts chemins
- Calcul des distances et d'un plus court chemin entre un sommet  $x_0$  et tous les autres sommets du graphe

## Principe de l'algorithme de Dijkstra

# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les **plus connus et efficaces** pour la recherche de distance et de plus courts chemins
- Calcul des distances et d'un **plus court chemin** entre un sommet  $x_0$  et tous les autres sommets du graphe
- Limitations : ne peut être utilisé que dans le cas où **toutes les valuations des arcs (ou arêtes) sont positives** (donc pas de circuits absorbants)

## Principe de l'algorithme de Dijkstra

# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les **plus connus et efficaces** pour la recherche de distance et de plus courts chemins
- Calcul des distances et d'un **plus court chemin** entre un sommet  $x_0$  et tous les autres sommets du graphe
- Limitations : ne peut être utilisé que dans le cas où **toutes les valuations des arcs (ou arêtes) sont positives** (donc pas de circuits absorbants)

## Principe de l'algorithme de Dijkstra

- Construction itérative d'un **ensemble de sommets marqués**, noté  $M$ 
  - Pour tout sommet marqué  $s$  : estimation  $d(s)$  = distance  $d(x_0, s)$
- À chaque étape

# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les **plus connus et efficaces** pour la recherche de distance et de plus courts chemins
- Calcul des distances et d'un **plus court chemin** entre un sommet  $x_0$  et tous les autres sommets du graphe
- Limitations : ne peut être utilisé que dans le cas où **toutes les valuations des arcs (ou arêtes) sont positives** (donc pas de circuits absorbants)

## Principe de l'algorithme de Dijkstra

- Construction itérative d'un **ensemble de sommets marqués**, noté  $M$   
→ Pour tout sommet marqué  $s$  : estimation  $d(s)$  = distance  $d(x_0, s)$
- À chaque étape
  - 1 Sélection d'un **sommet non marqué  $x$**  dont la **distance estimée  $d(x)$**  est la **plus petite** parmi les distances estimées des sommets non marqués

# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les **plus connus et efficaces** pour la recherche de distance et de plus courts chemins
- Calcul des distances et d'un **plus court chemin** entre un sommet  $x_0$  et tous les autres sommets du graphe
- Limitations : ne peut être utilisé que dans le cas où **toutes les valuations des arcs (ou arêtes) sont positives** (donc pas de circuits absorbants)

## Principe de l'algorithme de Dijkstra

- Construction itérative d'un **ensemble de sommets marqués**, noté  $M$ 
  - Pour tout sommet marqué  $s$  : estimation  $d(s)$  = distance  $d(x_0, s)$
- À chaque étape
  - 1 Sélection d'un **sommet non marqué  $x$**  dont la **distance estimée  $d(x)$**  est la **plus petite** parmi les distances estimées des sommets non marqués
  - 2 Ajout de  $x$  à l'ensemble  $M$



# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les **plus connus et efficaces** pour la recherche de distance et de plus courts chemins
- Calcul des distances et d'un **plus court chemin** entre un sommet  $x_0$  et tous les autres sommets du graphe
- Limitations : ne peut être utilisé que dans le cas où **toutes les valuations des arcs (ou arêtes) sont positives** (donc pas de circuits absorbants)

## Principe de l'algorithme de Dijkstra

- Construction itérative d'un **ensemble de sommets marqués**, noté  $M$   
→ Pour tout sommet marqué  $s$  : estimation  $d(s)$  = distance  $d(x_0, s)$
- À chaque étape
  - 1 Sélection d'un **sommet non marqué  $x$**  dont la **distance estimée  $d(x)$**  est la **plus petite** parmi les distances estimées des sommets non marqués
  - 2 Ajout de  $x$  à l'ensemble  $M$
  - 3 Mise à jour des **distances estimées des successeurs non marqués de  $x$**

# Algorithme de Dijkstra (1)

## Présentation de l'algorithme de Dijkstra

- Un des algorithmes les **plus connus et efficaces** pour la recherche de distance et de plus courts chemins
- Calcul des distances et d'un **plus court chemin** entre un sommet  $x_0$  et tous les autres sommets du graphe
- Limitations : ne peut être utilisé que dans le cas où **toutes les valuations des arcs (ou arêtes) sont positives** (donc pas de circuits absorbants)

## Principe de l'algorithme de Dijkstra

- Construction itérative d'un **ensemble de sommets marqués**, noté  $M$ 
  - Pour tout sommet marqué  $s$  : estimation  $d(s)$  = distance  $d(x_0, s)$
- À chaque étape
  - 1 Sélection d'un **sommet non marqué  $x$**  dont la **distance estimée  $d(x)$**  est la **plus petite** parmi les distances estimées des sommets non marqués
  - 2 Ajout de  $x$  à l'ensemble  $M$
  - 3 Mise à jour des **distances estimées des successeurs non marqués de  $x$**
  - À recommencer jusqu'à ce que tous les sommets soient marqués

# Algorithme de Dijkstra (2)

**Données : Graphe valué  $G = (S, A, v)$  sans valuations négatives ; sommet  $x_0$**

```
// Initialisation des tableaux  $d$  et  $P$  et de l'ensemble  $M$ 
1   $M = \emptyset$ ;                                     // Aucun sommet marqué
2  pour chaque  $s \in S$  faire
3       $P[s] = nul$ ;
4      si  $s = x_0$  alors
5           $d[s] = 0$ ;
6      fin si
7      sinon
8           $d[s] = +\infty$ ;
9      fin si
10 fin pour chaque

    // Boucle principale
11 tant que  $M \neq S$  faire
12      $x = \arg \min_{s \in S, s \notin M} d[s]$ ;           // Sommet non marqué de distance min.
13      $M = M \cup \{x\}$ ;                             // Ajout de  $x$  aux sommets marqués
14     pour chaque  $y \in \Gamma^+(x), y \notin M$  faire
15         si  $d[x] + v(x, y) < d[y]$  alors
16              $d[y] = d[x] + v(x, y)$ ;                 // Mise à jour de la distance
17              $P[y] = x$ ;                               // Mise à jour du père
18         fin si
19     fin pour chaque
20 fin tq
```

## Exemple (1)

Soit le graphe  $G$  dont les sommets sont nommés de  $A$  à  $D$  et défini par la matrice de valuation suivante :

$$G = \begin{pmatrix} +\infty & 8 & 6 & 2 \\ +\infty & +\infty & +\infty & +\infty \\ +\infty & 3 & +\infty & +\infty \\ +\infty & 5 & 1 & +\infty \end{pmatrix}$$

### 1 Initialisation de l'algorithme

$x$	$M$	$d[A] \quad P[A]$	$d[B] \quad P[B]$	$d[C] \quad P[C]$	$d[D] \quad P[D]$
$x_0 = A$	$\emptyset$	0 <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>

## Exemple (1)

Soit le graphe  $G$  dont les sommets sont nommés de  $A$  à  $D$  et défini par la matrice de valuation suivante :

$$G = \begin{pmatrix} +\infty & 8 & 6 & 2 \\ +\infty & +\infty & +\infty & +\infty \\ +\infty & 3 & +\infty & +\infty \\ +\infty & 5 & 1 & +\infty \end{pmatrix}$$

### 1 Initialisation de l'algorithme

$x$	$M$	$d[A] \quad P[A]$	$d[B] \quad P[B]$	$d[C] \quad P[C]$	$d[D] \quad P[D]$
$x_0 = A$	$\emptyset$	0 nul	$+\infty$ nul	$+\infty$ nul	$+\infty$ nul

### 2 Itération 1 : choix du sommet $x = A$ , de distance 0

$x$	$M$	$d[A] \quad P[A]$	$d[B] \quad P[B]$	$d[C] \quad P[C]$	$d[D] \quad P[D]$
$x_0 = A$	$\emptyset$	0 nul	$+\infty$ nul	$+\infty$ nul	$+\infty$ nul
$A$	$\{A\}$	—	$0 + 8 < +\infty ?$ oui donc : 8 <sub>A</sub>	$0 + 6 < +\infty ?$ oui donc : 6 <sub>A</sub>	$0 + 2 < +\infty ?$ oui donc : 2 <sub>A</sub>

## Exemple (2)

③ Itération 2 : choix du sommet  $x = D$ , de distance 2

$x$	$M$	$d[A] \quad P[A]$	$d[B] \quad P[B]$	$d[C] \quad P[C]$	$d[D] \quad P[D]$
$x_0 = A$	$\emptyset$	0 <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>
$A$	$\{A\}$	–	8 $_A$	6 $_A$	2 $_A$
$D$	$\{A, D\}$	–	2 + 5 < 8 ? oui donc : 7 $_D$	2 + 1 < 6 ? oui donc : 3 $_D$	–

## Exemple (2)

③ **Itération 2** : choix du sommet  $x = D$ , de distance 2

$x$	$M$	$d[A]_{P[A]}$	$d[B]_{P[B]}$	$d[C]_{P[C]}$	$d[D]_{P[D]}$
$x_0 = A$	$\emptyset$	0 nul	$+\infty$ nul	$+\infty$ nul	$+\infty$ nul
$A$	$\{A\}$	–	8 $_A$	6 $_A$	2 $_A$
$D$	$\{A, D\}$	–	2 + 5 < 8 ? oui donc : 7 $_D$	2 + 1 < 6 ? oui donc : 3 $_D$	–

④ **Itération 3** : choix du sommet  $x = C$ , de distance 3

$x$	$M$	$d[A]_{P[A]}$	$d[B]_{P[B]}$	$d[C]_{P[C]}$	$d[D]_{P[D]}$
$x_0 = A$	$\emptyset$	0 nul	$+\infty$ nul	$+\infty$ nul	$+\infty$ nul
$A$	$\{A\}$	–	8 $_A$	6 $_A$	2 $_A$
$D$	$\{A, D\}$	–	7 $_D$	3 $_D$	–
$C$	$\{A, D, C\}$	–	3 + 3 < 7 ? oui donc : 6 $_C$	–	–

## Exemple (2)

③ **Itération 2** : choix du sommet  $x = D$ , de distance 2

$x$	$M$	$d[A] \quad P[A]$	$d[B] \quad P[B]$	$d[C] \quad P[C]$	$d[D] \quad P[D]$
$x_0 = A$	$\emptyset$	0 <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>
$A$	$\{A\}$	–	8 $A$	6 $A$	2 $A$
$D$	$\{A, D\}$	–	2 + 5 < 8 ? oui donc : 7 $D$	2 + 1 < 6 ? oui donc : 3 $D$	–

④ **Itération 3** : choix du sommet  $x = C$ , de distance 3

$x$	$M$	$d[A] \quad P[A]$	$d[B] \quad P[B]$	$d[C] \quad P[C]$	$d[D] \quad P[D]$
$x_0 = A$	$\emptyset$	0 <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>
$A$	$\{A\}$	–	8 $A$	6 $A$	2 $A$
$D$	$\{A, D\}$	–	7 $D$	3 $D$	–
$C$	$\{A, D, C\}$	–	3 + 3 < 7 ? oui donc : 6 $C$	–	–

⑤ **Itération 4** : choix du sommet  $x = B$ , de distance 6

$x$	$M$	$d[A] \quad P[A]$	$d[B] \quad P[B]$	$d[C] \quad P[C]$	$d[D] \quad P[D]$
$x_0 = A$	$\emptyset$	0 <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>	$+\infty$ <i>nul</i>
$A$	$\{A\}$	–	8 $A$	6 $A$	2 $A$
$D$	$\{A, D\}$	–	7 $D$	3 $D$	–
$C$	$\{A, D, C\}$	–	6 $C$	–	–
$B$	$\{A, D, C, B\}$	0 <i>nul</i>	6 $C$	3 $D$	2 $A$



# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$

## Principe de la construction du plus court chemin du sommet $s$

# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
- On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$

## Principe de la construction du plus court chemin du sommet $s$

# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
  - On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$
- Utilisation du tableau des pères pour construire les plus courts chemins

## Principe de la construction du plus court chemin du sommet $s$

# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
  - On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$
- Utilisation du tableau des pères pour construire les plus courts chemins

## Principe de la construction du plus court chemin du sommet $s$

- Construction itérative du plus court chemin de  $x_0$  à  $s$ , à partir de la fin

# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
  - On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$
- Utilisation du tableau des pères pour construire les plus courts chemins

## Principe de la construction du plus court chemin du sommet $s$

- Construction itérative du plus court chemin de  $x_0$  à  $s$ , à partir de la fin
- Initialisation
  - ▶ Initialisation du sommet courant  $x$  avec le sommet  $s$  :  $x \leftarrow s$
  - ▶ Initialiation du plus court chemin de  $x_0$  à  $s$  avec le sommet  $x$  :  $c \leftarrow [x]$

# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
  - On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$
- Utilisation du tableau des pères pour construire les plus courts chemins

## Principe de la construction du plus court chemin du sommet $s$

- Construction itérative du plus court chemin de  $x_0$  à  $s$ , à partir de la fin
- Initialisation
  - ▶ Initialisation du sommet courant  $x$  avec le sommet  $s$  :  $x \leftarrow s$
  - ▶ Initialiation du plus court chemin de  $x_0$  à  $s$  avec le sommet  $x$  :  $c \leftarrow [x]$
- À chaque étape

# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
  - On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$
- Utilisation du tableau des pères pour construire les plus courts chemins

## Principe de la construction du plus court chemin du sommet $s$

- Construction itérative du plus court chemin de  $x_0$  à  $s$ , à partir de la fin
- Initialisation
  - ▶ Initialisation du sommet courant  $x$  avec le sommet  $s$  :  $x \leftarrow s$
  - ▶ Initialiation du plus court chemin de  $x_0$  à  $s$  avec le sommet  $x$  :  $c \leftarrow [x]$
- À chaque étape
  - 1 Sélection du nouveau sommet courant  $x$  en choisissant son père :  $x \leftarrow P[x]$

# Construction des plus courts chemins

## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
  - On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$
- Utilisation du tableau des pères pour construire les plus courts chemins

## Principe de la construction du plus court chemin du sommet $s$

- Construction itérative du plus court chemin de  $x_0$  à  $s$ , à partir de la fin
- Initialisation
  - ▶ Initialisation du sommet courant  $x$  avec le sommet  $s$  :  $x \leftarrow s$
  - ▶ Initialiation du plus court chemin de  $x_0$  à  $s$  avec le sommet  $x$  :  $c \leftarrow [x]$
- À chaque étape
  - 1 Sélection du nouveau sommet courant  $x$  en choisissant son père :  $x \leftarrow P[x]$
  - 2 Ajout de  $x$  au début du plus court chemin :  $c \leftarrow x + c$



# Construction des plus courts chemins

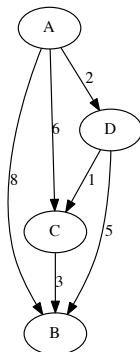
## Présentation de la construction des plus courts chemins

- À la fin de l'algo. de Dijkstra : distances calculées, de  $x_0$  à tout sommet  $s$
- On peut déterminer un plus court chemin de  $x_0$  à  $s$ , pour tout sommet  $s$
- Utilisation du tableau des pères pour construire les plus courts chemins

## Principe de la construction du plus court chemin du sommet $s$

- Construction itérative du plus court chemin de  $x_0$  à  $s$ , à partir de la fin
- Initialisation
  - ▶ Initialisation du sommet courant  $x$  avec le sommet  $s$  :  $x \leftarrow s$
  - ▶ Initialiation du plus court chemin de  $x_0$  à  $s$  avec le sommet  $x$  :  $c \leftarrow [x]$
- À chaque étape
  - 1 Sélection du nouveau sommet courant  $x$  en choisissant son père :  $x \leftarrow P[x]$
  - 2 Ajout de  $x$  au début du plus court chemin :  $c \leftarrow x + c$- À recommencer jusqu'à ce que le sommet courant soit le sommet  $x_0$

# Exemple



$P[A]$	$P[B]$	$P[C]$	$P[D]$
nul	$C$	$D$	$A$

- Un plus court chemin de  $A$  à  $B$  est :  $A \rightarrow D \rightarrow C \rightarrow B$

# Propriétés

## Propriété

A toutes les étapes de l'algorithme, chaque sommet  $s$  marqué vérifie  
 $d(s) = d(x_0, s)$

## Propriété

Après  $i$  itérations de la boucle principale de l'algorithme, ensemble des sommets marqués :  $M = \{x_0, x_1, \dots, x_{i-1}\}$

→ Cela implique : à la fin de l'algorithme, on a bien calculé les distances de  $x_0$  à  $s$