Recherche de plus courts chemins

Jean Cousty

Graphes et algorithmes

Plan du cours

1. Plus courts chemins
2. Algorithme de Dijkstra
3. Algorithme de recherche arrière

Réseau

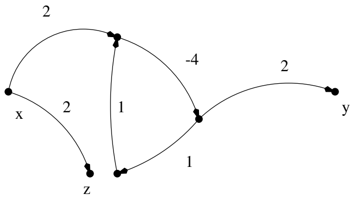
Définition

- *Une* réseau *est un triplet* $R = (E, \Gamma, \ell)$ *tel que*
  - (E, Γ) *est un graphe sans boucle; et*
  - ℓ *est une application de* $\overrightarrow{\Gamma}$ *dans* $\mathbb{R}$
- *Si* (E, Γ, ℓ) *est un réseau et si* $u \in \overrightarrow{\Gamma}$ *est un arc, le nombre réel* $\ell(u)$ *est appelé la* longueur de u

Réseau

Définition

- *Une* réseau *est un triplet* $R = (E, \Gamma, \ell)$ *tel que*
  - (E, Γ) *est un graphe sans boucle; et*
  - ℓ *est une application de* $\overrightarrow{\Gamma}$ *dans* $\mathbb{R}$
- *Si* (E, Γ, ℓ) *est un réseau et si* $u \in \overrightarrow{\Gamma}$ *est un arc, le nombre réel* $\ell(u)$ *est appelé la* longueur de *u*



Notations

- Dans la suite de ce cours, $N = (E, \Gamma, \ell)$ désigne toujours un réseau et G désigne le graphe $G = (E, \Gamma)$
- Si $u = (x, y)$ est un arc de G, nous écrirons $\ell(x, y)$ au lieu de $\ell((x, y))$

Longueur d'un chemin

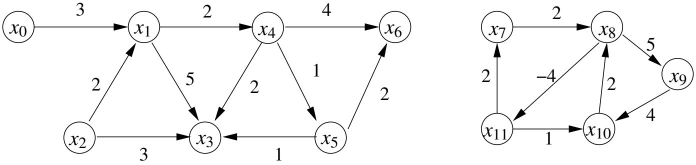
Définition

- *Soit* $\pi = (x_0, \ldots, x_n)$ *un chemin dans* G
- *La* longueur de π (dans N) *est la somme des longueurs des arcs qui apparaissent dans* π*:*
  - $L(\pi) = \sum\{\ell(x_i, x_{i+1}) \mid 0 \leq i \leq n-1\}$

Longueur d'un chemin

Définition

- *Soit $\pi = (x_0, \ldots, x_n)$ un chemin dans G*
- *La* longueur de π (dans N) *est la somme des longueurs des arcs qui apparaissent dans π:*
  - $L(\pi) = \sum\{\ell(x_i, x_{i+1}) \mid 0 \leq i \leq n-1\}$



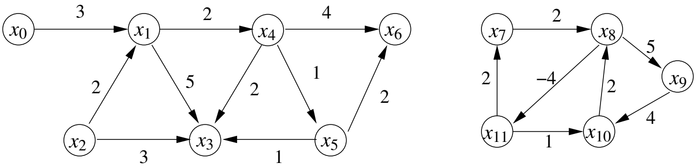
$L((x_0, x_1, x_3)) = 8$

Plus court chemin

Définition

- *Soient x et y deux sommets dans E*
- *Un* plus court chemin de x à y (dans N) *est un chemin π de x à y dont la longueur est inférieure ou égale à la longueur de tout autre chemin de x à y:*
  - *pour tout chemin π' de x à y, nous avons $L(\pi) \leq L(\pi')$*

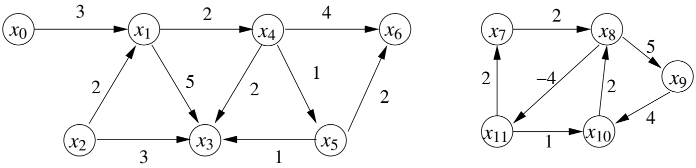
Plus courts chemins: illustration



Example

- $\pi = (x_0, x_1, x_3)$

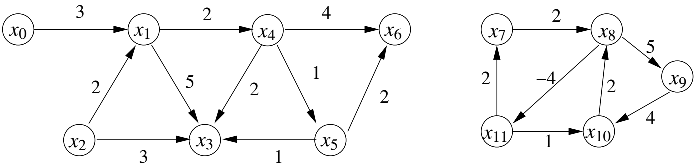
Plus courts chemins: illustration



Example

- $\pi = (x_0, x_1, x_3)$ *n'est pas un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 8$*)*

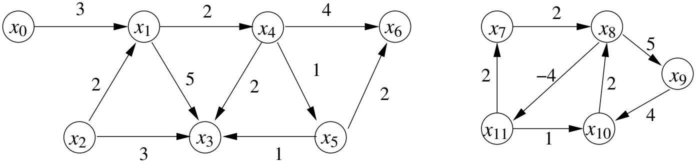
Plus courts chemins: illustration



Example

- $\pi = (x_0, x_1, x_3)$ *n'est pas un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 8$*)*
- $\pi = (x_0, x_1, x_4, x_3)$ *est un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 7$*)*

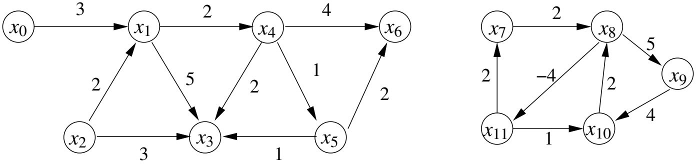
Plus courts chemins: illustration



Example

- $\pi = (x_0, x_1, x_3)$ *n'est pas un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 8$*)*
- $\pi = (x_0, x_1, x_4, x_3)$ *est un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 7$*)*
- *plus court chemin de* x_2 *à* x_0 *?*

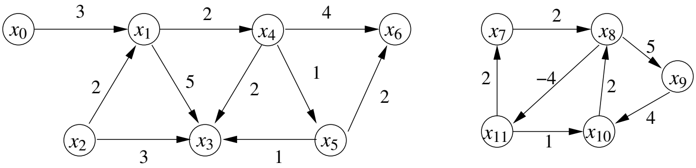
Plus courts chemins: illustration



Example

- $\pi = (x_0, x_1, x_3)$ *n'est pas un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 8$*)*
- $\pi = (x_0, x_1, x_4, x_3)$ *est un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 7$*)*
- *Il n'y a pas de plus court chemin de* x_2 *à* x_0

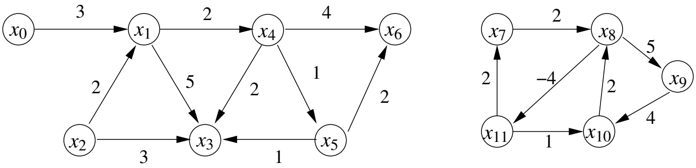
Plus courts chemins: illustration



Example

- $\pi = (x_0, x_1, x_3)$ *n'est pas un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 8$*)*
- $\pi = (x_0, x_1, x_4, x_3)$ *est un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 7$*)*
- *Il n'y a pas de plus court chemin de* x_2 *à* x_0
- *plus court chemin de* x_7 *à* x_9 *?*

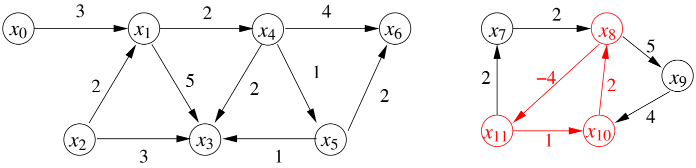
Plus courts chemins: illustration



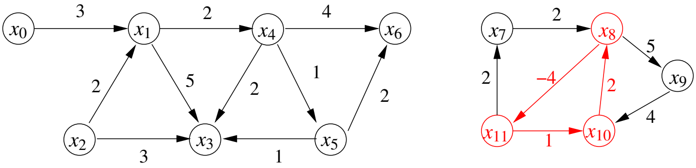
Example

- $\pi = (x_0, x_1, x_3)$ *n'est pas un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 8$*)*
- $\pi = (x_0, x_1, x_4, x_3)$ *est un plus court chemin de* x_0 *à* x_3 *(*$L(\pi) = 7$*)*
- *Il n'y a pas de plus court chemin de* x_2 *à* x_0
- *Il n'y a pas de plus court chemin de* x_7 *à* x_9

Circuit absorbant



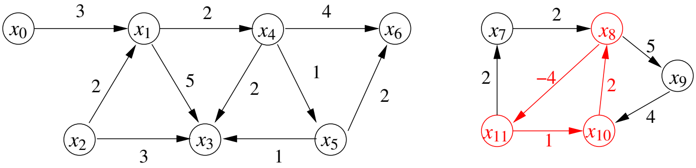
Circuit absorbant



Définition

- *Un* circuit absorbant *dans N est un circuit de longueur strictement négative*

Circuit absorbant



Définition

- *Un* circuit absorbant *dans* N *est un circuit de longueur strictement négative*

Remark. S'il y a un circuit absorbant dans (le graphe induit par) une composante fortement connexe de G, alors, quels que soient deux sommets de cette composante, il n'existe pas de plus court chemin de l'un à l'autre

Existance et unicité d'un plus court chemin

Propriété

- *Soit x un sommet de G*
- *Il existe une plus court chemin de x à tout autre sommet de G si et seulement si*
  - *pour tout sommet y de G, il existe un chemin de x à y*
  - *il n'y a pas de circuit absorbant dans N*

Existance et unicité d'un plus court chemin

Propriété

- *Soit x un sommet de G*
- *Il existe une plus court chemin de x à tout autre sommet de G si et seulement si*
  - *pour tout sommet y de G, il existe un chemin de x à y*
  - *il n'y a pas de circuit absorbant dans N*

Remarque

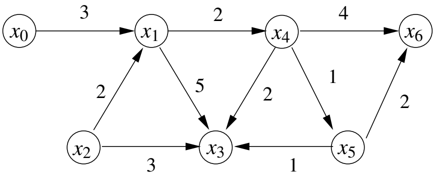
- *Étant données deux sommets x et y dans E, il peut exister plusieurs plus courts chemins différents de x à y*

Plus court chemin ou circuit absorbant

- Il existe des algorithmes pour
  1. trouver des plus courts chemins s'ils existent; et
  2. détecter si un graphe à un circuit négatif
- Exemple : algorithme de Bellman, algorithme de Floyd (voir TD)

Réseaux à longueurs positives

- Un *réseau à longueurs positives* est un réseau (E, Γ, ℓ) tel que:
  - $\forall u \in \overrightarrow{\Gamma}$, $\ell(u) \geq 0$

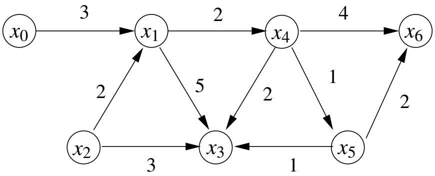


Réseaux à longueurs positives

- Un *réseau à longueurs positives* est un réseau (E, Γ, ℓ) tel que:
  - $\forall u \in \overrightarrow{\Gamma}, \ell(u) \geq 0$

Propriété

- *Si (E, Γ, ℓ) est un réseau à longueurs positives, alos pour tous éléments x et y de E*
  - *$\exists$ un chemin de x à y $\Leftrightarrow$ $\exists$ un plus court chemin from x à y*

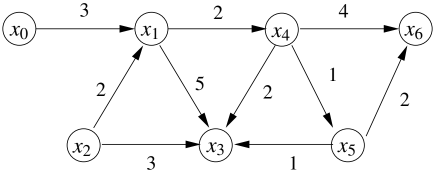


Longueurs des plus courts chemins

- Soit $N = (E, \Gamma, \ell)$ un réseau à longueurs positives, soit x un sommet dans E
- Nous définissons l'application $L_x : E \to \mathbb{R} \cup \{\infty\}$ par:

$$L_x(y) = \begin{cases} \text{la longueur d'un plus court chemin de } x \text{ à } y, \\ \qquad\qquad\qquad\qquad \text{si un tel chemin existe} \\ \infty \text{ , sinon} \end{cases}$$

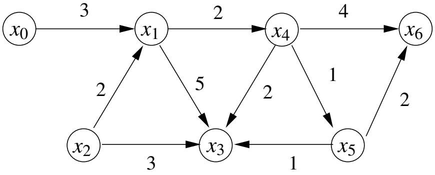
Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$							

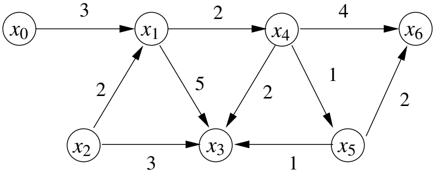
Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$	0						

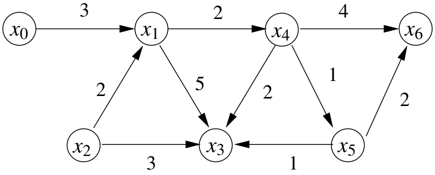
Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$	0	3					

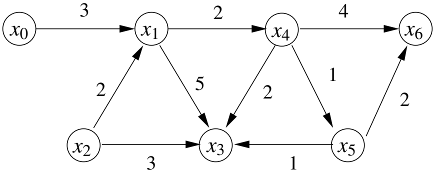
Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$	0	3	∞				

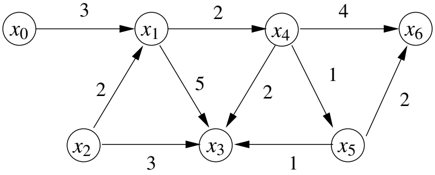
Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$	0	3	∞	7			

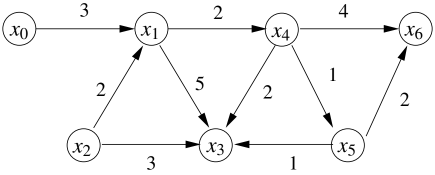
Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$	0	3	∞	7	5		

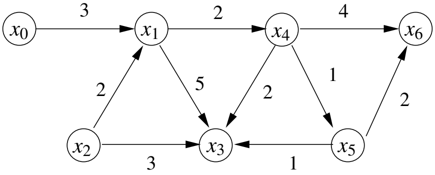
Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$	0	3	∞	7	5	6	

Illustration: application L_x



Example

$y =$	x_0	x_1	x_2	x_3	x_4	x_5	x_6
$L_{x_0}(y) =$	0	3	∞	7	5	6	8

Problèmes

1. Étant donné un réseau (E, Γ, ℓ) et deux sommets x et y dans E
   - Trouver un plus court chemin de x à y
   - Trouver la longueur $L_x(y)$ d'un plus court chemin de x à y
2. Étant donné un réseau (E, Γ, ℓ) et un sommet x dans E
   - Trouver pour chaque sommet y dans E la longueur $L_x(y)$ d'un plus court chemin de x à y
3. Étant donné un réseau (E, Γ, ℓ)
   - Trouver, pour chaque paire x, y de sommets dans E, la longueur d'un plus court chemin de x à y
4. Ayant résolu le problème 2
   - Résoudre le problème 1

Algorithme de Dijkstra

1. Étant donné un réseau (E, Γ, ℓ) et deux sommets x et y dans E
   - Trouver un plus court chemin de x à y
   - Trouver la longueur $L_x(y)$ d'un plus court chemin de x à y
2. Étant donné un réseau (E, Γ, ℓ) et un sommet x dans E
   - Trouver pour chaque sommet y dans E la longueur $L_x(y)$ d'un plus court chemin de x à y
3. Étant donné un réseau (E, Γ, ℓ)
   - Trouver, pour chaque paire x, y de sommets dans E, la longueur d'un plus court chemin de x à y
4. Ayant résolu le problème 2
   - Résoudre le problème 1

Remarque

Dans l'algorithme de Dijkstta on suppose que le réseau N est à longueurs positives

Longueur des plus courts chemins depuis un sommet donné

Algorithme de DIJKSTRA (Donnée: (E, Γ, ℓ), $n = |E|$, $x \in E$; **Résultat**: L_x)

$\overline{S} := \emptyset$;

Pour chaque $y \in E$ **Faire** $L_x[y] = \infty$; $\overline{S} := \overline{S} \cup \{y\}$;

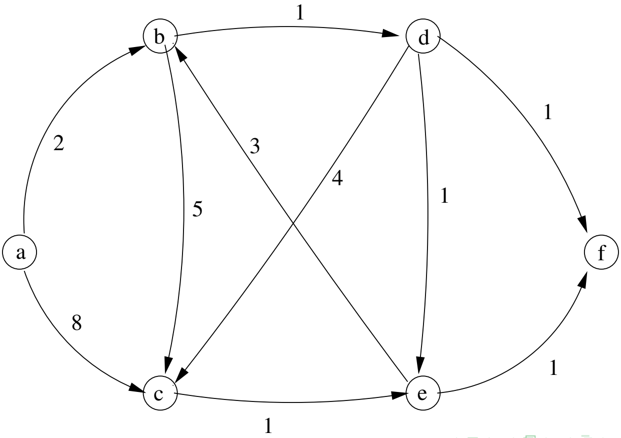
$L_x[x] := 0$; $k := 0$; $\mu := 0$;

Tant que $k < n$ et $\mu \neq \infty$ **Faire**

- Trouver un sommet $y^\star \in \overline{S}$ tel que $L_x[y^\star] = \min\{L_x[y], y \in \overline{S}\}$
- Supprimer $y^\star$ de $\overline{S}$
- $k++$; $\mu := L_x[y^\star]$;
- **Pour chaque** $y \in \Gamma(y^\star) \cap \overline{S}$ **Faire**
  - $L_x[y] := \min\{L_x[y], L_x[y^\star] + \ell(y^\star, y)\}$;

Longueur des plus courts chemins depuis un sommet donné

- *Exercice.* Executer "à la main" l'algorithme de Dijsktra sur le réseau suivant avec $x = a$, puis sur n'imprte quel réseau à longueurs positives de votre choix



Invariant de boucle de l'algorithme de Dijkstra (# 1)

- Soient $x \in E$ et $\mu \in \mathbb{R}$
- Un sous-ensemble S de E est *μ-separant (pour x)* si les deux propositions suivantes sont vraies:

Invariant de boucle de l'algorithme de Dijkstra (# 1)

- Soient $x \in E$ et $\mu \in \mathbb{R}$
- Un sous-ensemble S de E est *μ-separant (pour x)* si les deux propositions suivantes sont vraies:
  1. S contient tout sommet y tel que la longueur $L_x(y)$ d'un plus court chemin de x à y est strictement inférieur à μ

Invariant de boucle de l'algorithme de Dijkstra (# 1)

- Soient $x \in E$ et $\mu \in \mathbb{R}$
- Un sous-ensemble S de E est *μ-separant (pour x)* si les deux propositions suivantes sont vraies:
  1. S contient tout sommet y tel que la longueur $L_x(y)$ d'un plus court chemin de x à y est strictement inférieur à μ
  2. $\overline{S} = E \setminus S$ contient tout sommet y tel que la longueur d'un plus court chemins de x à y est strictement supérieure à μ

Invariant de boucle de l'algorithme de Dijkstra (# 2)

- Soient $x \in E$, $\mu \in \mathbb{R}$ et soit S un ensemble μ-séparant pour x
- Un *S-chemin* est un chemin dont les sommets intermediaires appartiennent tous à S

Invariant de boucle de l'algorithme de Dijkstra (# 2)

- Soient $x \in E$, $\mu \in \mathbb{R}$ et soit S un ensemble μ-séparant pour x
- Un *S-chemin* est un chemin dont les sommets intermediaires appartiennent tous à S
- La longueur d'un plus court S-chemin de x à y est désignée par $L_x^S(y)$

Invariant de boucle de l'algorithme de Dijkstra (# 2)

- Soient $x \in E$, $\mu \in \mathbb{R}$ et soit S un ensemble μ-séparant pour x
- Un *S-chemin* est un chemin dont les sommets intermediaires appartiennent tous à S
- La longueur d'un plus court S-chemin de x à y est désignée par $L_x^S(y)$

Propriété (preuve de l'algorithme de Dijkstra)

- *Soit* $y^\star \in \overline{S}$ *tel que* $L_x^S(y^\star) = \min\{L_x^s(y) \mid y \in \overline{S}\}$

Invariant de boucle de l'algorithme de Dijkstra (# 2)

- Soient $x \in E$, $\mu \in \mathbb{R}$ et soit S un ensemble μ-séparant pour x
- Un *S-chemin* est un chemin dont les sommets intermediaires appartiennent tous à S
- La longueur d'un plus court S-chemin de x à y est désignée par $L_x^S(y)$

Propriété (preuve de l'algorithme de Dijkstra)

- *Soit* $y^\star \in \overline{S}$ *tel que* $L_x^S(y^\star) = \min\{L_x^s(y) \mid y \in \overline{S}\}$
- *Alors,* $L_x^S(y^\star) = L_x(y^\star)$

Invariant de boucle de l'algorithme de Dijkstra (# 2)

- Soient $x \in E$, $\mu \in \mathbb{R}$ et soit S un ensemble μ-séparant pour x
- Un *S-chemin* est un chemin dont les sommets intermediaires appartiennent tous à S
- La longueur d'un plus court S-chemin de x à y est désignée par $L_x^S(y)$

Propriété (preuve de l'algorithme de Dijkstra)

- *Soit* $y^\star \in \overline{S}$ *tel que* $L_x^S(y^\star) = \min\{L_x^s(y) \mid y \in \overline{S}\}$
- *Alors,* $L_x^S(y^\star) = L_x(y^\star)$
- *Alors,* $S \cup \{y^\star\}$ *est un ensemble* μ'*-séparant avec* $\mu' = L_x^S(y^\star)$

Longueur des plus courts chemins depuis un sommet donné

Algorithme de DIJKSTRA (Donnée: (E, Γ, ℓ), $n = |E|$, $x \in E$; **Résultat**: L_x)

$\overline{S} := \emptyset$;

Pour chaque $y \in E$ **Faire** $L_x[y] = \infty$; $\overline{S} := \overline{S} \cup \{y\}$;

$L_x[x] := 0$; $k := 0$; $\mu := 0$;

Tant que $k < n$ and $\mu \neq \infty$ **Faire**

- Trouver un sommet $y^\star \in \overline{S}$ such that $L_x[y^\star] = \min\{L_x[y], y \in \overline{S}\}$
- Supprimer $y^\star$ de $\overline{S}$
- $k++$; $\mu := L_x[y^\star]$;
- **Pour chaque** $y \in \Gamma(y^\star) \cap \overline{S}$ **Faire**
  - $L_x[y] := \min\{L_x[y], L_x[y^\star] + \ell(y^\star, y)\}$;

Analyse de compelxité

Complexity

- *Initialisation:* $O(n)$
- *Boucle* **Tant que** *(ligne 4):* $O(n)$
- *Trouver/supprimer (line 5,6):* $O(n^2)$
- *Boucle* **Pour chaque** *(ligne 7):* $O(n+m)$
- *Algorithme de DIJKSTRA:* $O(n^2)$

Analyse de compelxité

Complexity

- *Initialisation:* $O(n)$
- *Boucle* **Tant que** *(ligne 4):* $O(n)$
- *Trouver/supprimer (line 5,6):* $O(n^2)$
- *Boucle* **Pour chaque** *(ligne 7):* $O(n+m)$
- *Algorithme de DIJKSTRA:* $O(n^2)$
- *peut être réduit à* $O(n\log(n)+m)$

Exercice: recherche arrière d'un plus court chemin

- Proposer un algorithme dont les **données** sont :
  - un réseau N à longueurs positives
  - une paire (x, y) de sommets
- et dont le **resultat** est :
  - un plus court chemin de x à y si un tel chemin existe

Aide. Commencer par calculer la longueur $L_x(z)$ pour tout sommet z dans E en appelant l'algorithme de Disjkstra à la manière d'un appel de procédure, puis commencer la “recherche arrière”.