Recherche Opérationelle
Alternance

Moritz Mühlenthaler, Zoltán Szigeti

moritz.muehlenthaler@grenoble-inp.fr

G-SCOP (Sciences pour la Conception, l'Optimisation et la Production de Grenoble)
Équipe OC (Optimisation Combinatoire)

Plus Courts Chemins

Motivation

Modélisation

- Réseau : graphe orienté pondéré G, c
  - $G = (V, A)$
  - $c : A \to \mathrm{R}$
- c-coût $c(P)$ d'un chemin $P = v_0 e_1 v_1 \dots e_k v_k$

$$c(P) := \sum_{1 \leq i \leq k} c(e_i)$$

- distance : $\mathrm{dist}_{G,c}(s, t) = c$-coût minimum d'un s-t chemin.

Modélisation

- Réseau : graphe orienté pondéré G, c
  - $G = (V, A)$
  - $c : A \to \mathrm{R}$
- c-coût $c(P)$ d'un chemin $P = v_0 e_1 v_1 \dots e_k v_k$

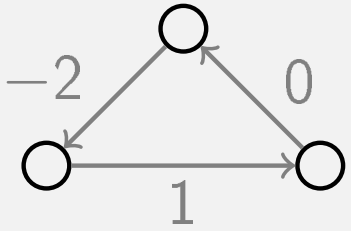
$$c(P) := \sum_{1 \leq i \leq k} c(e_i)$$

- distance : $\mathrm{dist}_{G,c}(s, t) = c$-coût minimum d'un s-t chemin.

Problème : Étant donné un réseau G, c et un sommet s de G, trouver pour tout sommet t un s-t chemin de c-coût minimum.

Circuits Absorbants

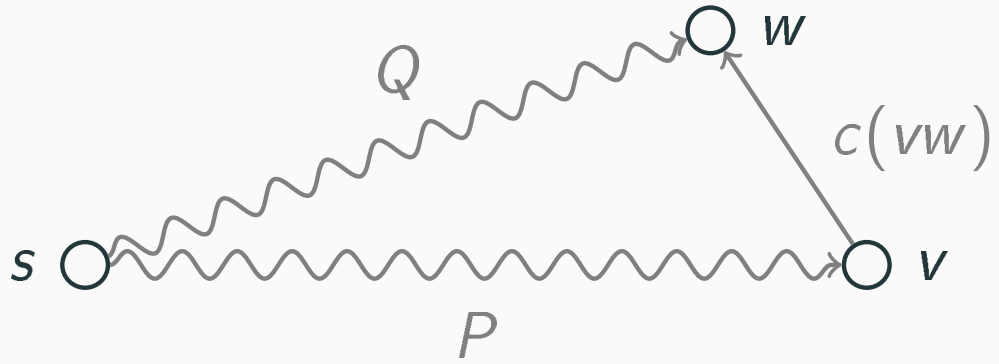
Circuit absorbant : circuit C de c-coût négatif ; $c(C) < 0$.



- avec circuit absorbant
  - chemin : une solution optimale n'existe pas (pourquoi ?)
  - chemin élémentaire : problème est aussi difficile que trouver un s-t chemin hamiltonien
- sans circuit absorbant
  - un chemin optimale est élémentaire (pourquoi ?)
  - on peut résoudre le problème

Algorithmes : Aperçu

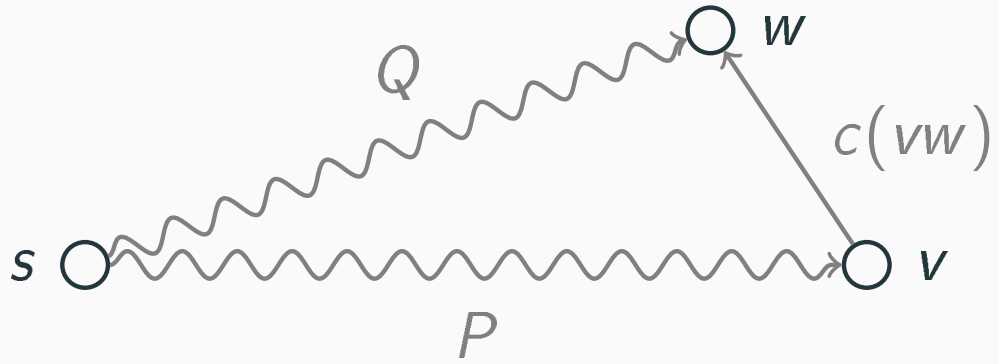
- Idée fondamentale



Supposons que vw est un arc et on connaît un s-v chemin P et un s-w chemin Q. Si $c(P) + c(vw) < c(Q)$, alors on a trouvé un plus court s-w chemin $P + vw$.

Algorithmes : Aperçu

- Idée fondamentale



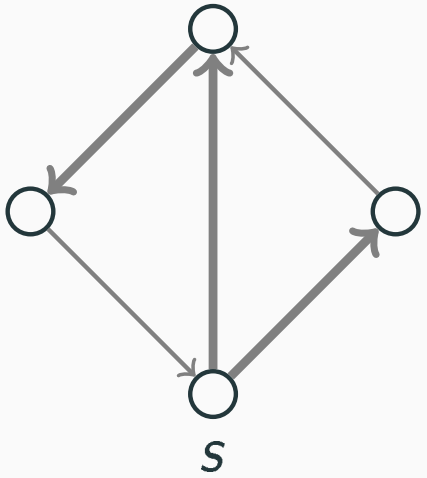
  Supposons que vw est un arc et on connaît un s-v chemin P et un s-w chemin Q. Si $c(P) + c(vw) < c(Q)$, alors on a trouvé un plus court s-w chemin $P + vw$.

- Cas spéciaux

G **sans circuit**	algorithme de Bellman
c **positif**	algorithme de Dijkstra
G **sans circuit absorbant**	algorithme de Bellman-Ford

Solutions Optimales

- Nous supposons dans ce qui suit que chaque sommet est atteignable de s.
- s-arborescence de G : graphe partiel F der G, tel qu'il existe un chemin unique dans F de s à chaque sommet.



- On peut supposer qu'une solution optimale (un ensemble de plus court chemins de s à chaque sommet) est une s-arborescence. (pourquoi ?)

Préparation : la procedure Scan

Algorithme 1 : Procedure Scan

Entrée : Réseau $G = (V, A)$, $c : A \to \mathbb{R}$, sommet $v \in V$, longueur y_w d'un s-w chemin pour tout $w \in V$, prédécesseur p_w de w pour tout $w \in V - s$

Sortie : valeurs y et prédécesseurs p mis-à-jour

1 **pour** *tout arc sortant* vw *de* v **faire**
2 **si** $y_w > y_v + c(vw)$ **alors**
3 $y_w \leftarrow y_v + c(vw)$
4 $p_w \leftarrow v$
5 **fin**

Réseaux sans Circuit : Algo de Bellman

Algorithme 2 : Algo de Bellman

Entrée : Réseau sans circuit $G = (V, A)$, $c : A \to \mathbb{R}$, sommet $s \in V$

Sortie : s-arborescence F

1 $y_s \leftarrow 0$ et $y_v \leftarrow \infty$ pour $v \in V - s$

2 $p_s \leftarrow 0$ et $p_v \leftarrow -1$ pour $v \in V - s$

3 **pour** $v \in V$ *en tri topologique* **faire**

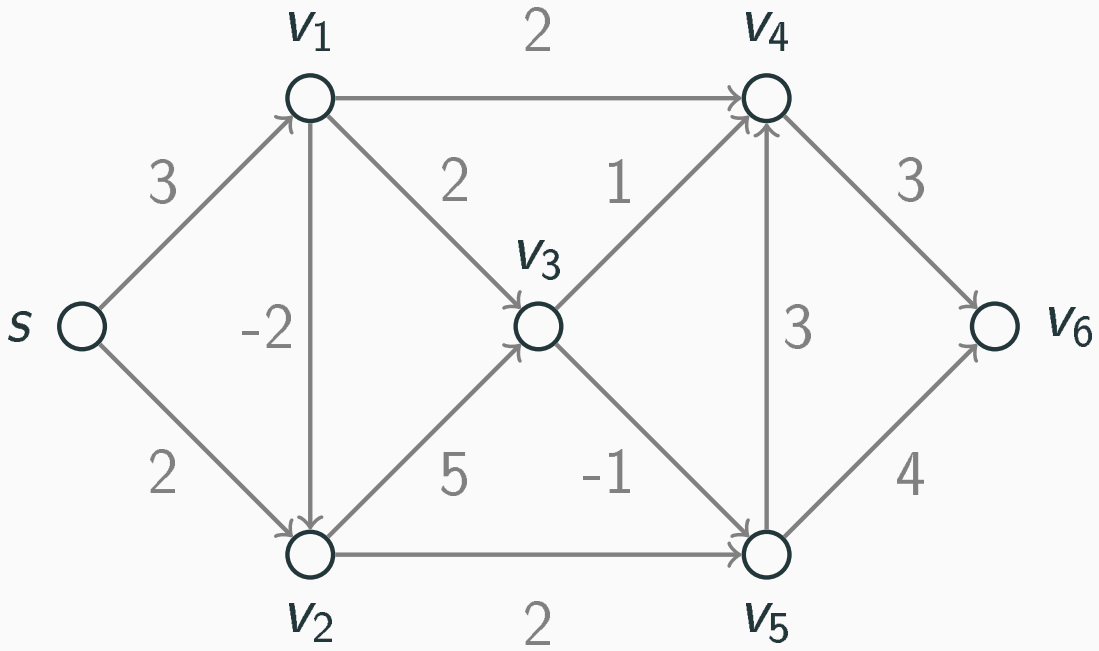
4 Scan(v)

5 $A_F \leftarrow \{p_v v \mid v \in V - s\}$

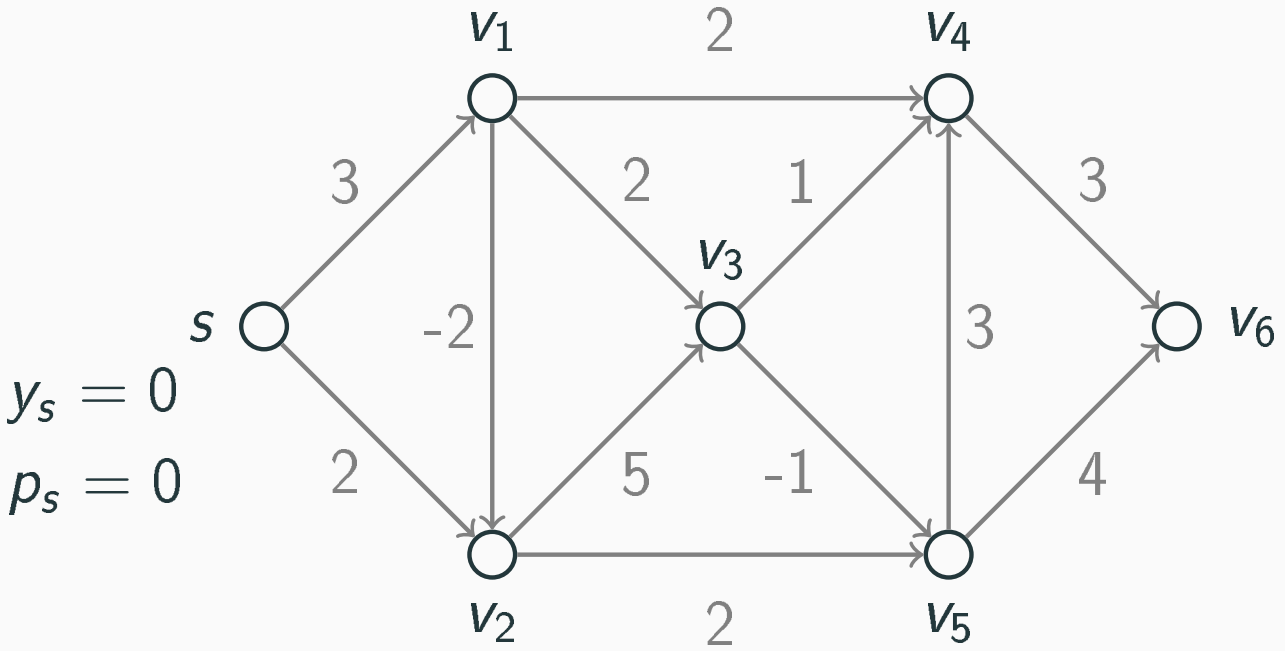
6 $F \leftarrow (V, A_F)$

7 **fin**

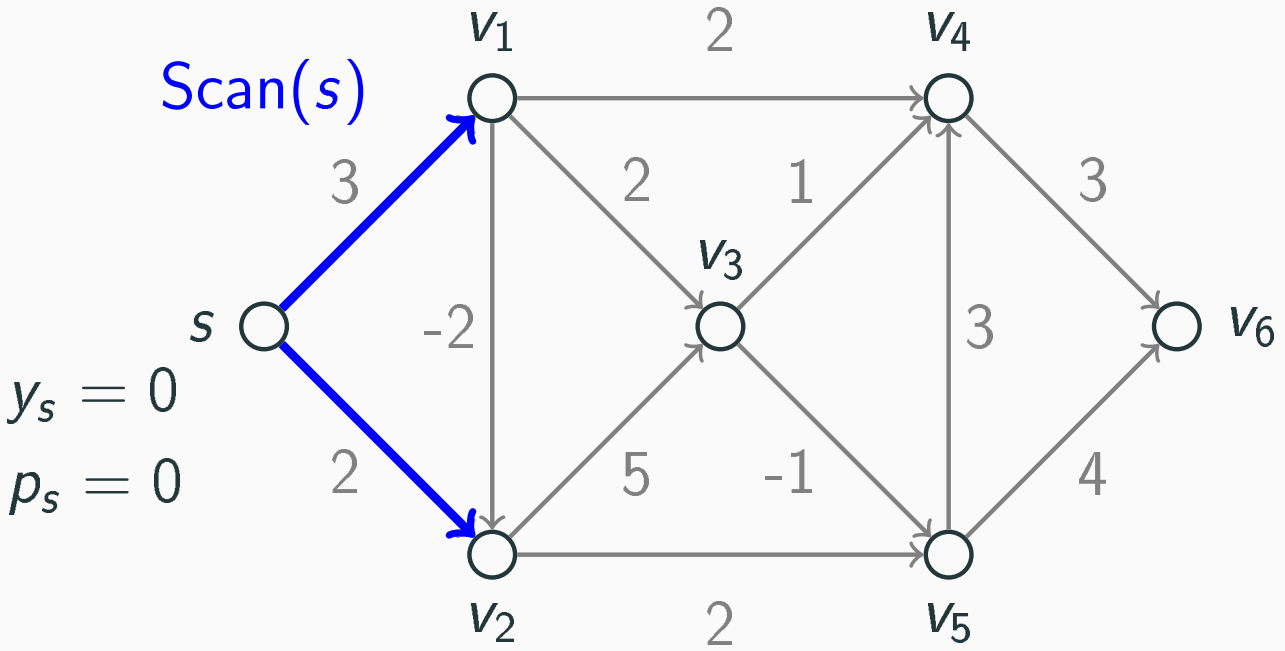
Exécution de l'Algo de Bellman



Exécution de l'Algo de Bellman



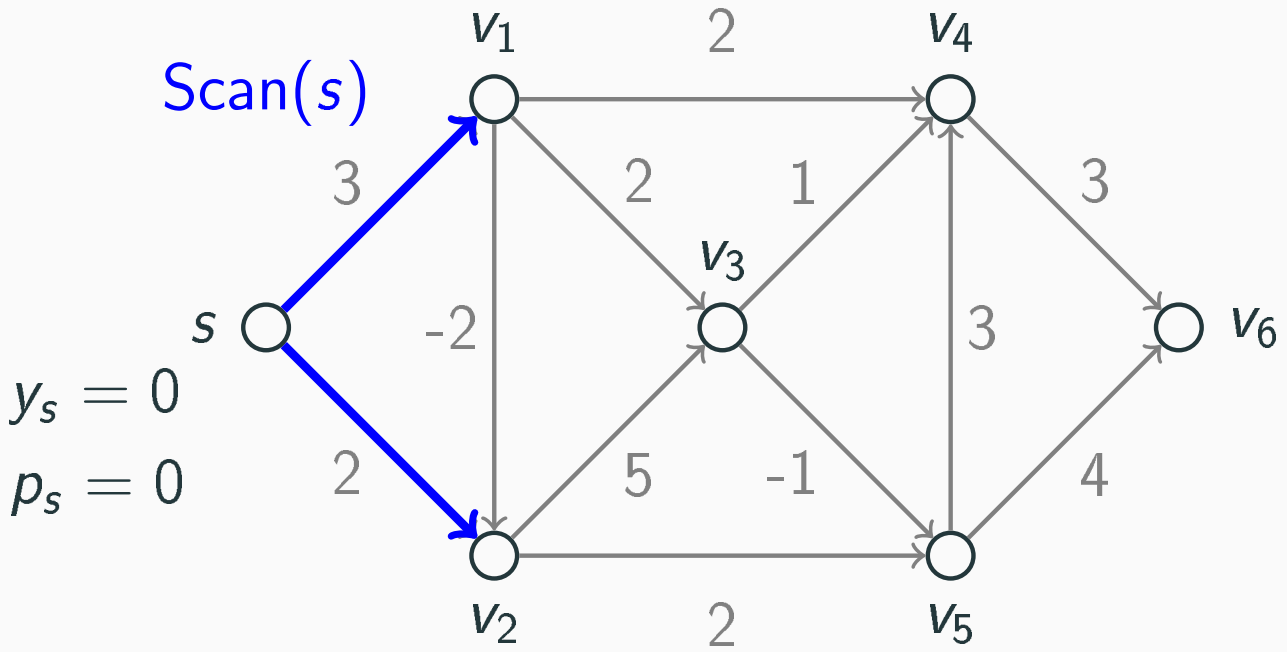
Exécution de l'Algo de Bellman



Exécution de l'Algo de Bellman

$y_{v_1} = 3$

$p_{v_1} = s$



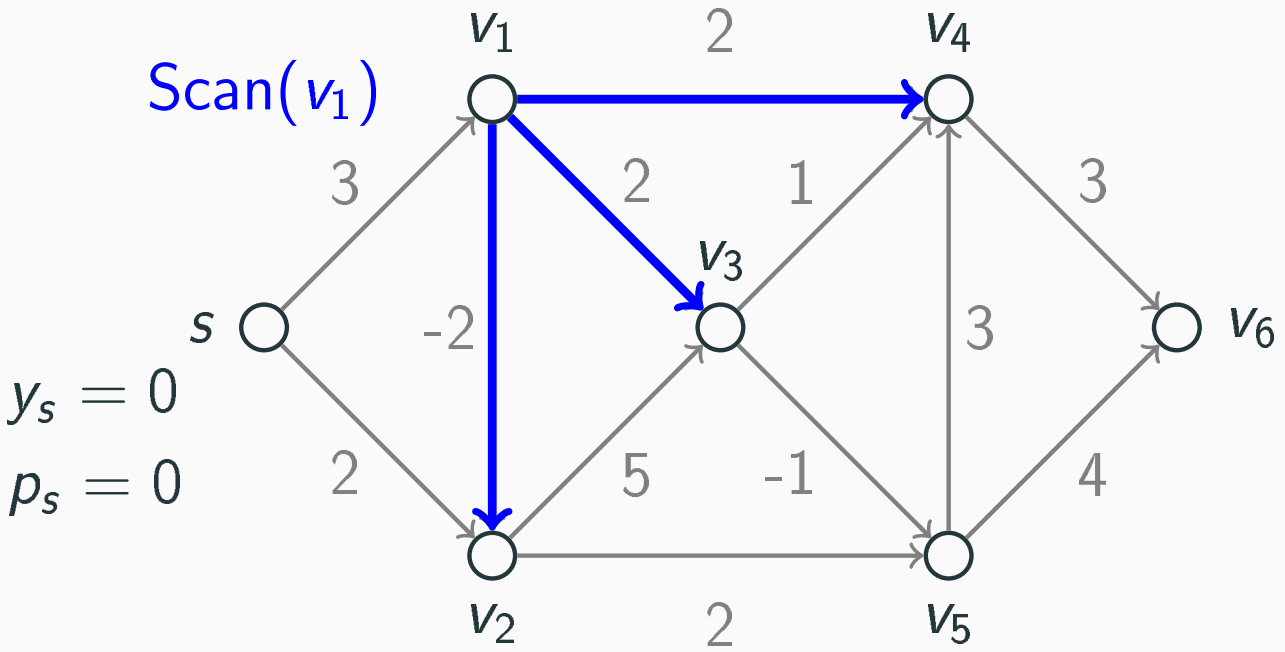
$y_{v_2} = 2$

$p_{v_2} = s$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$

$p_{v_1} = s$



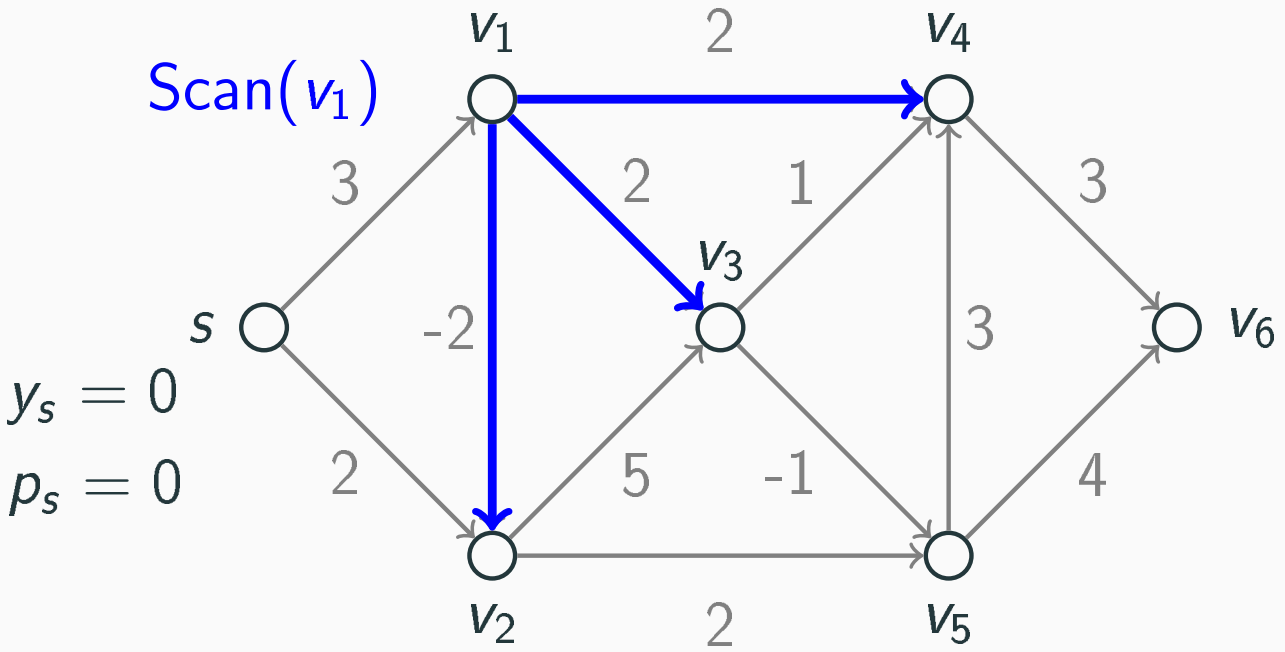
$y_{v_2} = 2$

$p_{v_2} = s$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



$y_s = 0$

$p_s = 0$

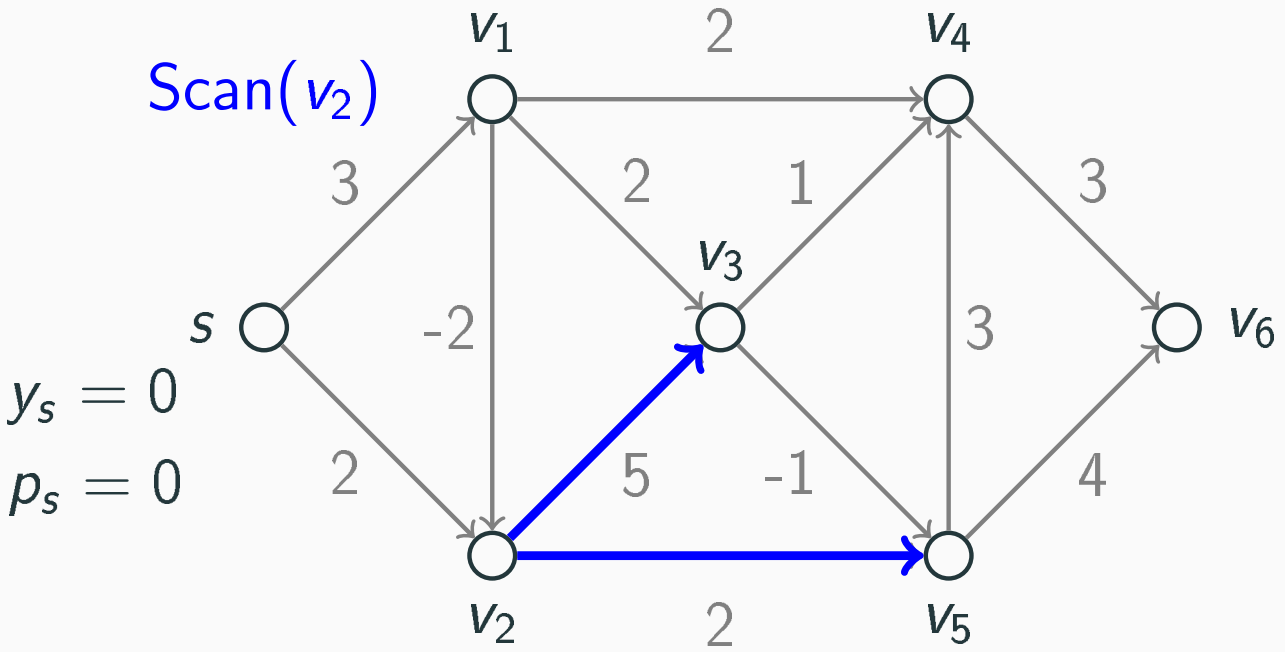
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



$y_s = 0$

$p_s = 0$

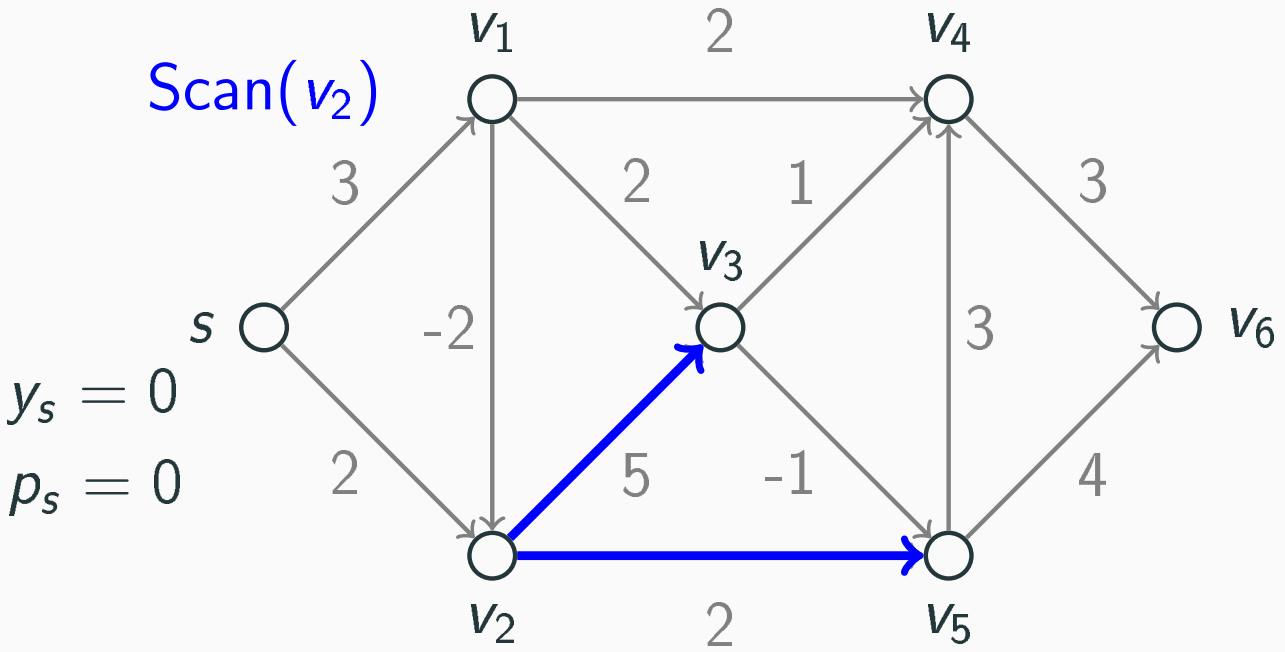
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



$y_s = 0$

$p_s = 0$

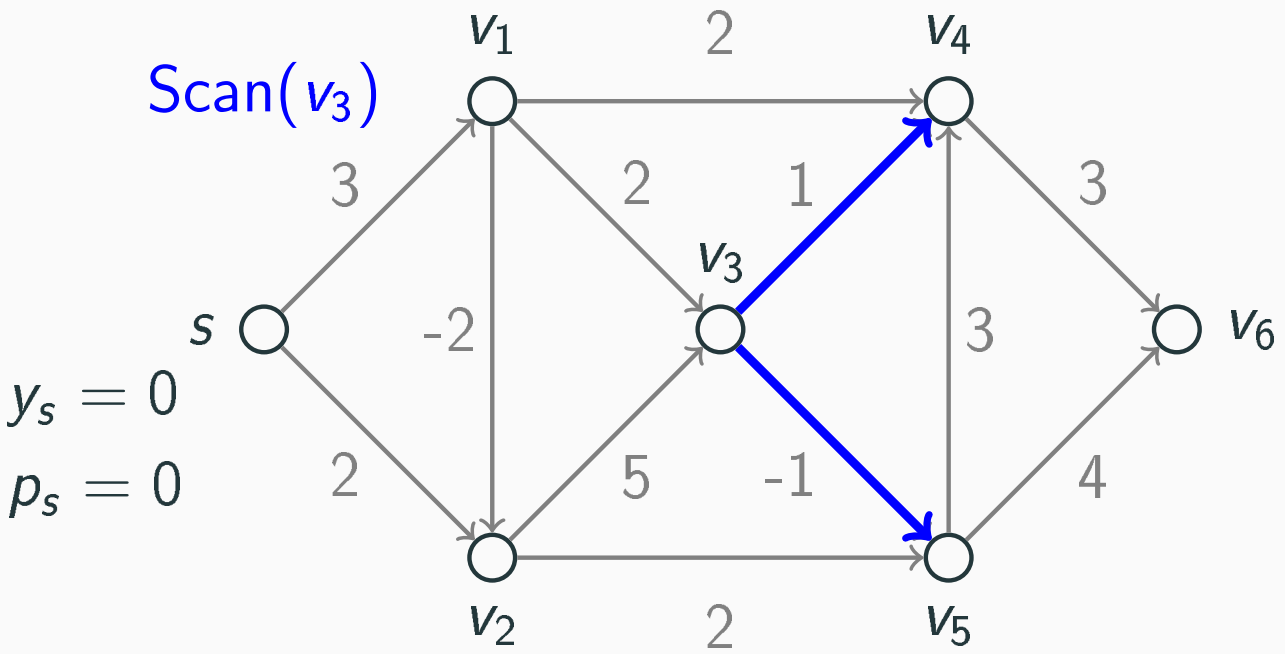
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$ $\quad$ $y_{v_5} = 3$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$ $\quad$ $p_{v_5} = v_2$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



$y_s = 0$

$p_s = 0$

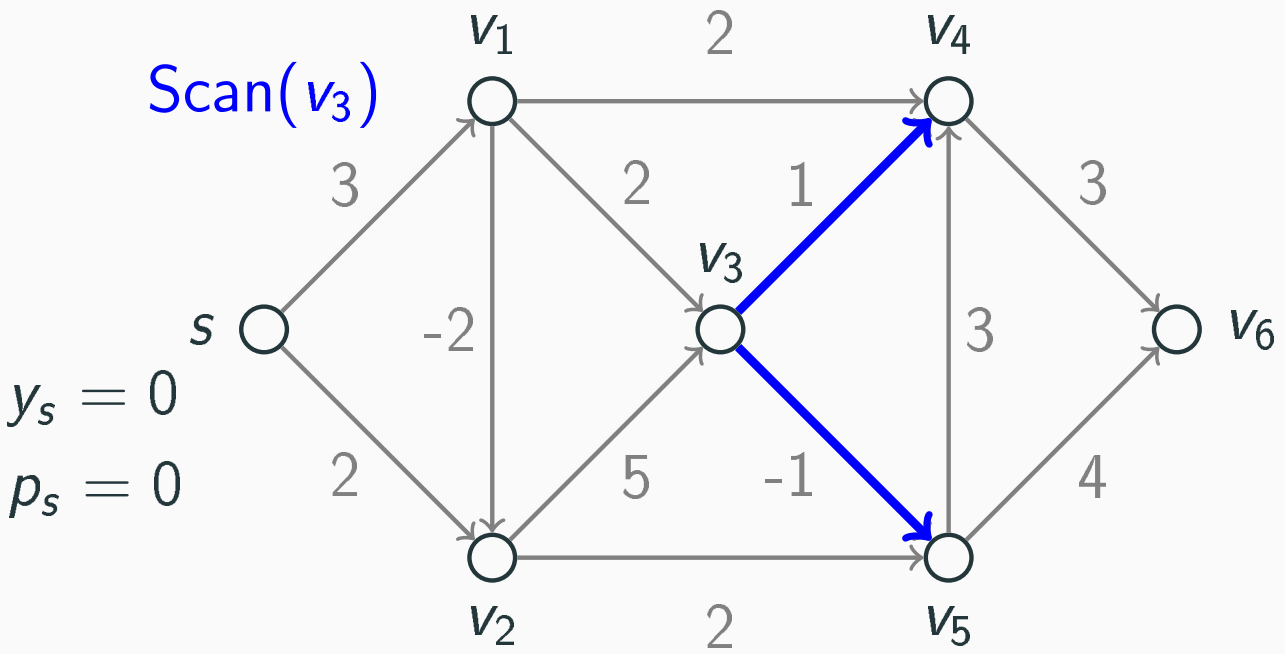
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$ $\quad$ $y_{v_5} = 3$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$ $\quad$ $p_{v_5} = v_2$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



$y_s = 0$

$p_s = 0$

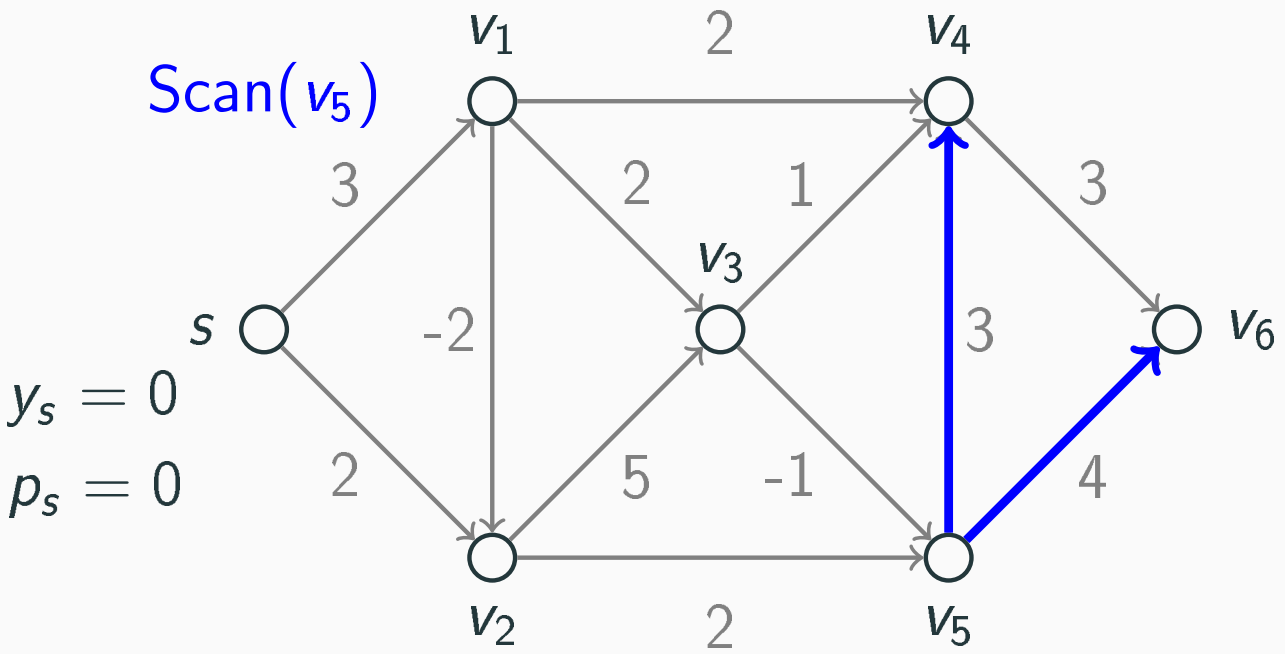
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$ $\quad$ $y_{v_5} = 3$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$ $\quad$ $p_{v_5} = v_2$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



$y_s = 0$

$p_s = 0$

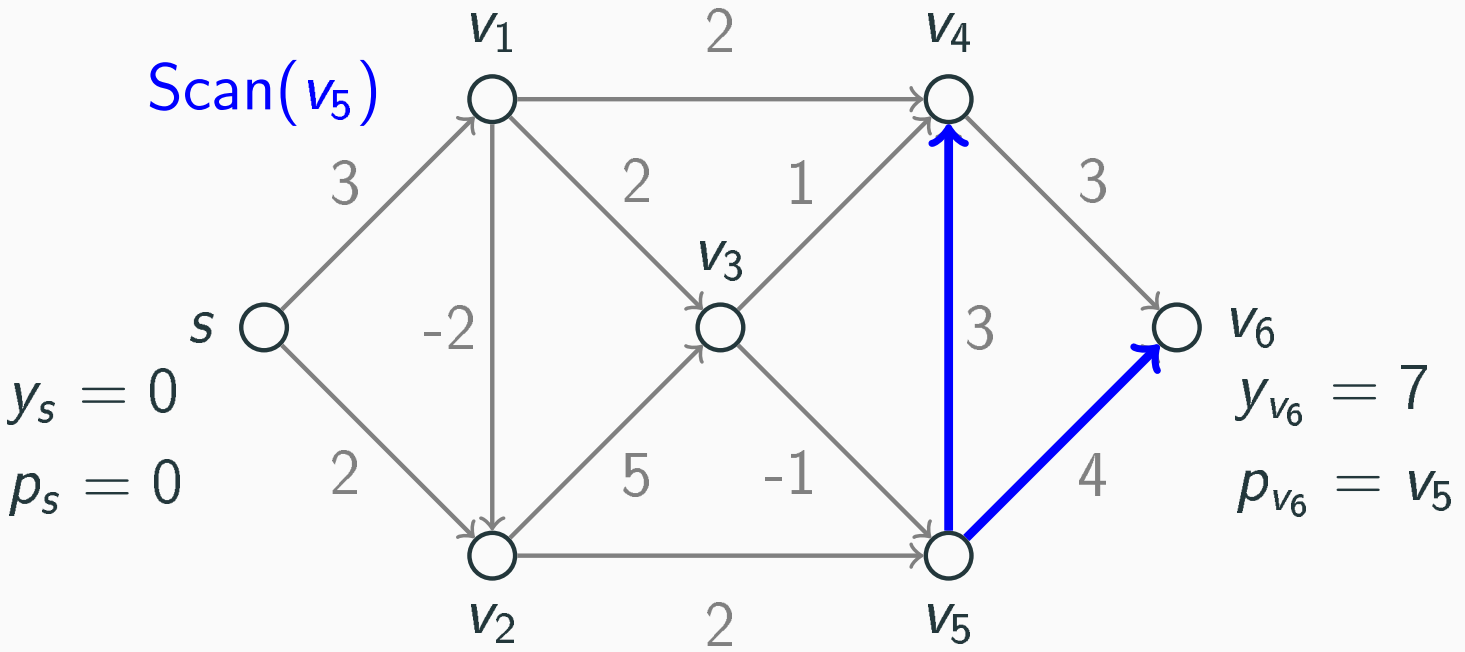
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$ $\quad$ $y_{v_5} = 3$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$ $\quad$ $p_{v_5} = v_2$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



$y_s = 0$

$p_s = 0$

$y_{v_6} = 7$

$p_{v_6} = v_5$

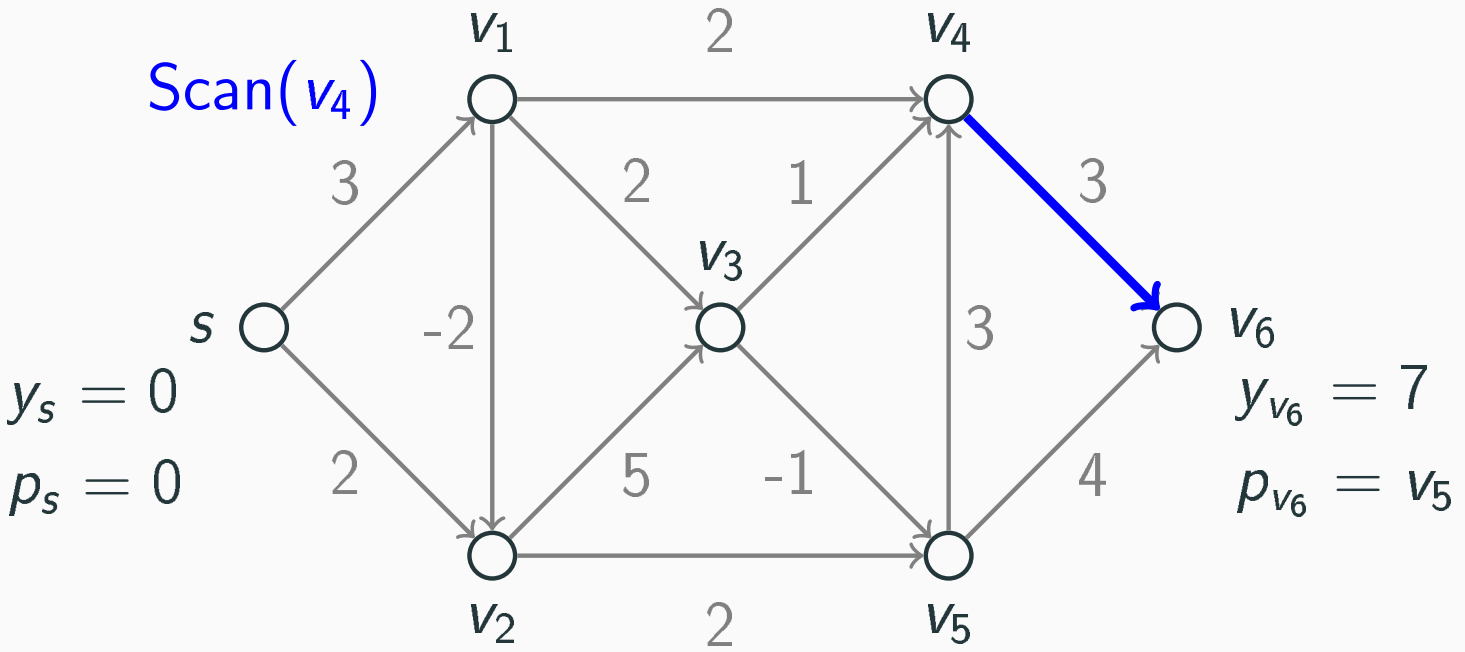
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$ $\quad$ $y_{v_5} = 3$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$ $\quad$ $p_{v_5} = v_2$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $p_{v_1} = s$

$y_{v_4} = 5$ $\quad$ $p_{v_4} = v_1$



$y_{v_3} = 5$ $\quad$ $p_{v_3} = v_1$

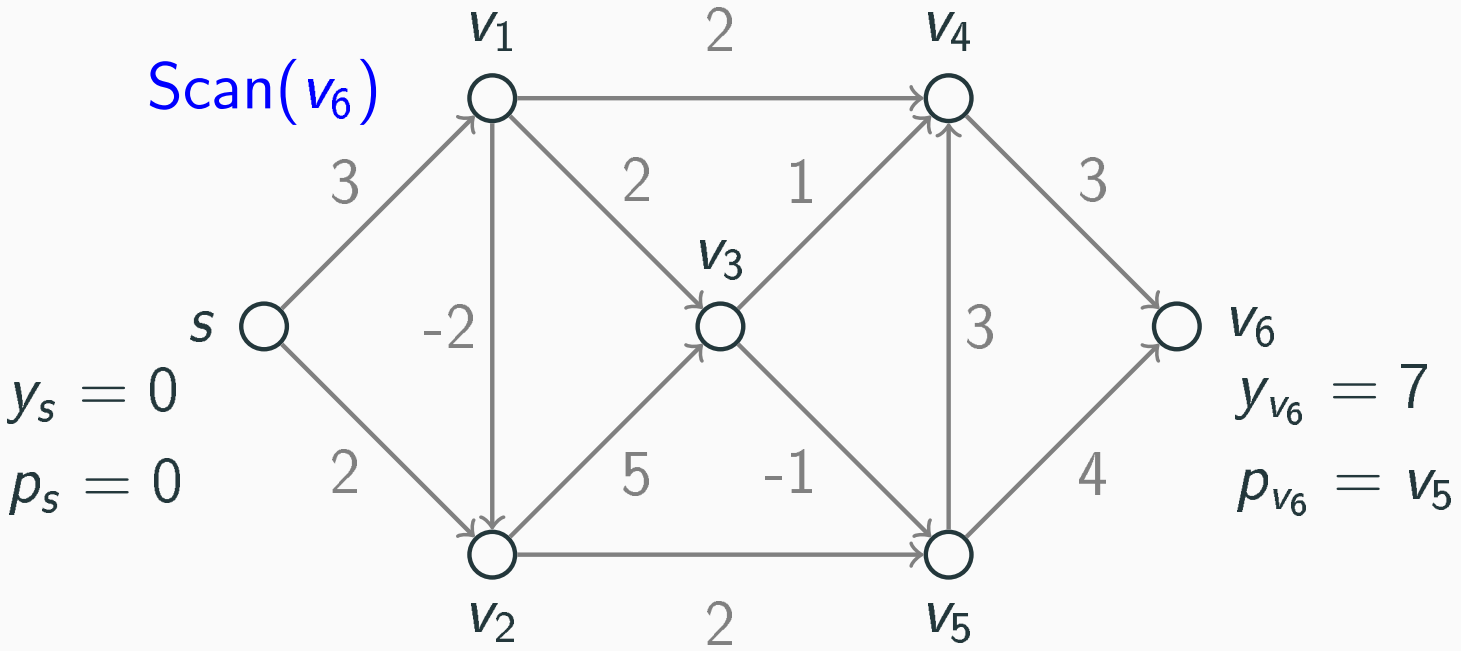
$y_{v_2} = 1$ $\quad$ $p_{v_2} = v_1$

$y_{v_5} = 3$ $\quad$ $p_{v_5} = v_2$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



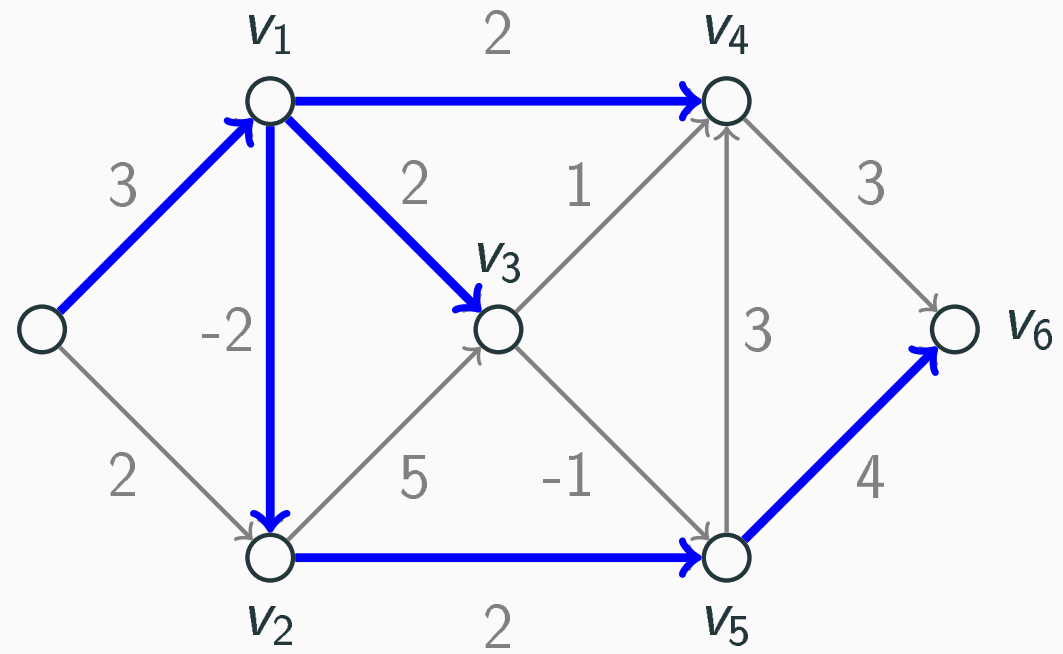
$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$ $\quad$ $y_{v_5} = 3$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$ $\quad$ $p_{v_5} = v_2$

Exécution de l'Algo de Bellman

$y_{v_1} = 3$ $\quad$ $y_{v_4} = 5$

$p_{v_1} = s$ $\quad$ $p_{v_4} = v_1$



s-arborescence

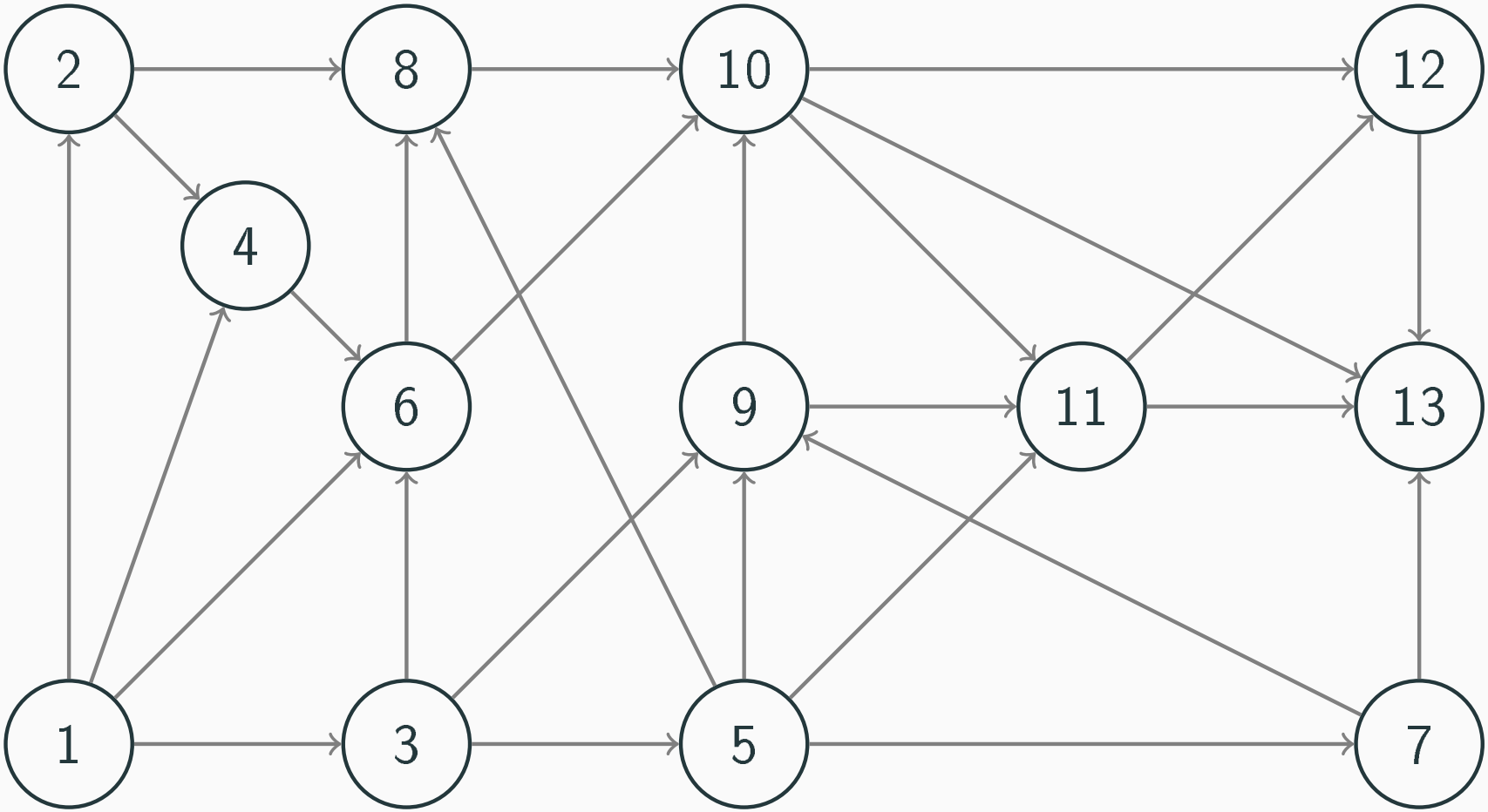
$y_s = 0$

$p_s = 0$

$y_{v_3} = 5$ $\quad$ $y_{v_2} = 1$ $\quad$ $y_{v_5} = 3$

$p_{v_3} = v_1$ $\quad$ $p_{v_2} = v_1$ $\quad$ $p_{v_5} = v_2$

Exercice



Exercice (cont.)

Sachant que les arcs horizontaux coûtent 2 chacun, les verticaux coûtent -1 et les obliques coûtent 4

1. Proposer un tri topologique des sommets.
2. En utilisant l'algorithme de Bellman, déterminer le plus *court* chemin de 1 à 13 et le plus *long* chemin de 1 à 13.

Réseaux avec Coûts Positifs : Algo de Dijkstra

Algorithme 3 : Dijkstra

Entrée : Réseau $G = (V, A)$, $c : A \to \mathbb{R}_{\geq 0}$, $s \in V$

Sortie : s-arborescence F

1 $y_s \leftarrow 0$ et $y_v \leftarrow \infty$ pour $v \in V - s$

2 $p_s \leftarrow 0$ et $p_v \leftarrow -1$ pour $v \in V - s$

3 $S \leftarrow \emptyset$

4 **tant que** $S \neq V$ **faire**

5 $\quad v \leftarrow \arg\min\{y_u \mid u \notin S\}$

6 $\quad S \leftarrow S \cup \{v\}$

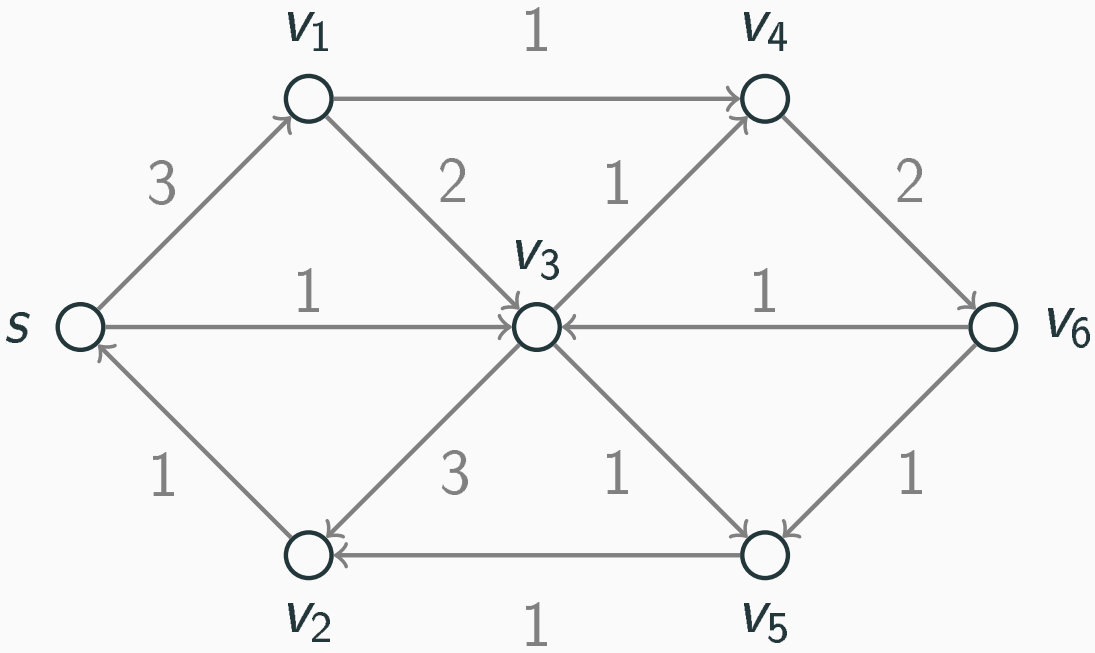
7 $\quad \text{Scan}(v)$

8 $A_F \leftarrow \{p_v v \mid v \in V - s\}$

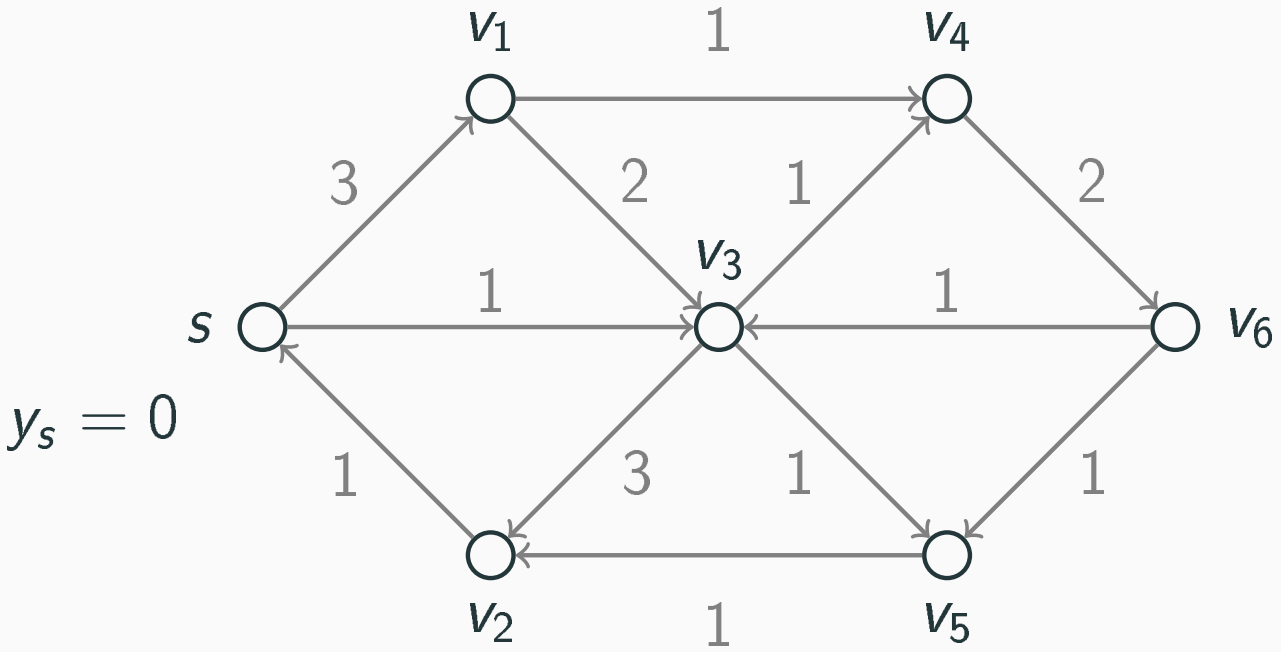
9 $F \leftarrow (V, A_F)$

10 **fin**

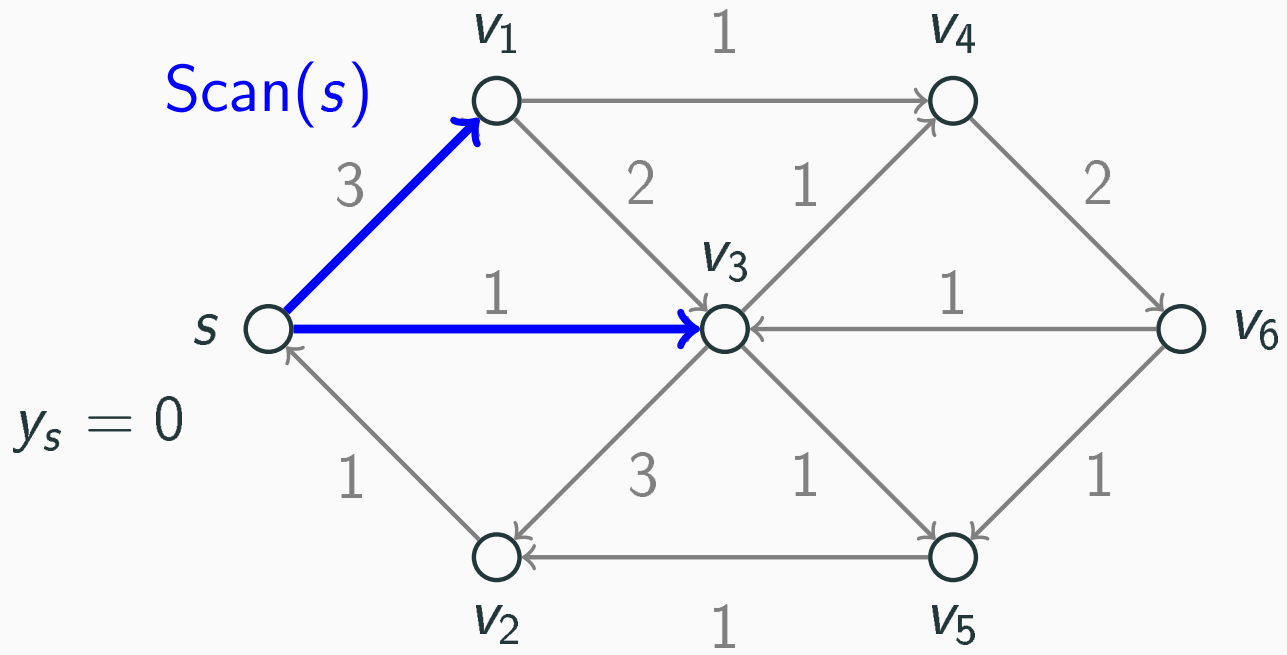
Exécution de l'Algo de Dijkstra



Exécution de l'Algo de Dijkstra



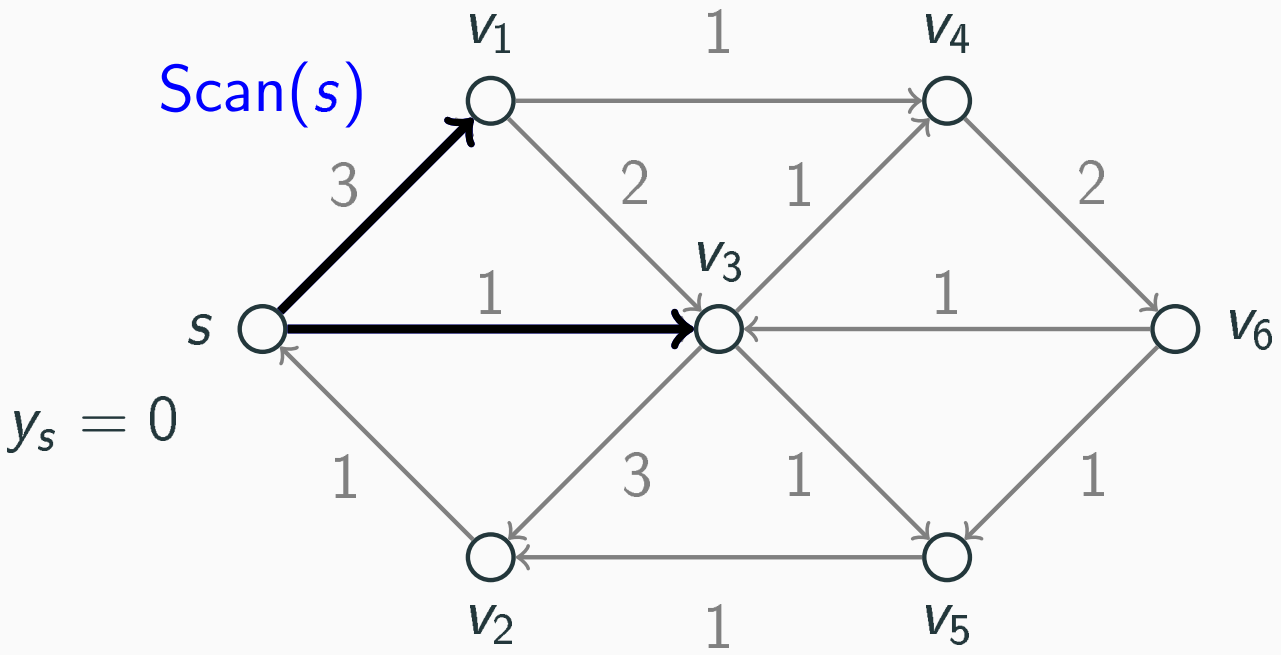
Exécution de l'Algo de Dijkstra



Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

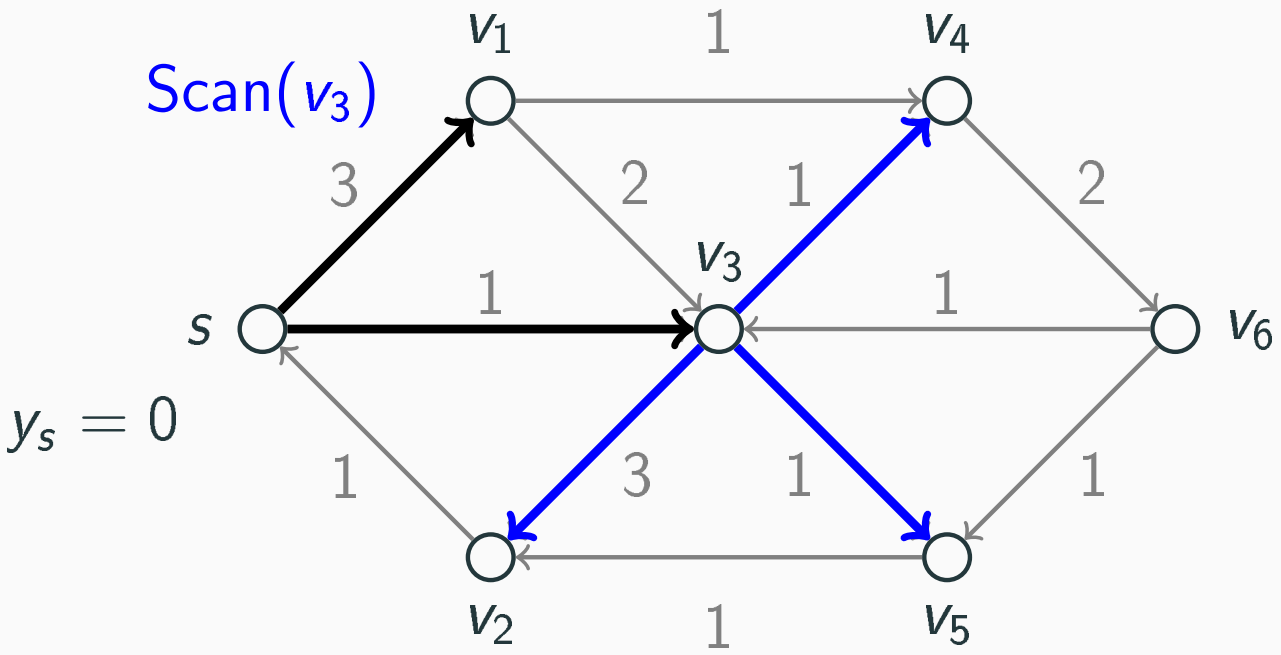
$y_{v3} = 1$



Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

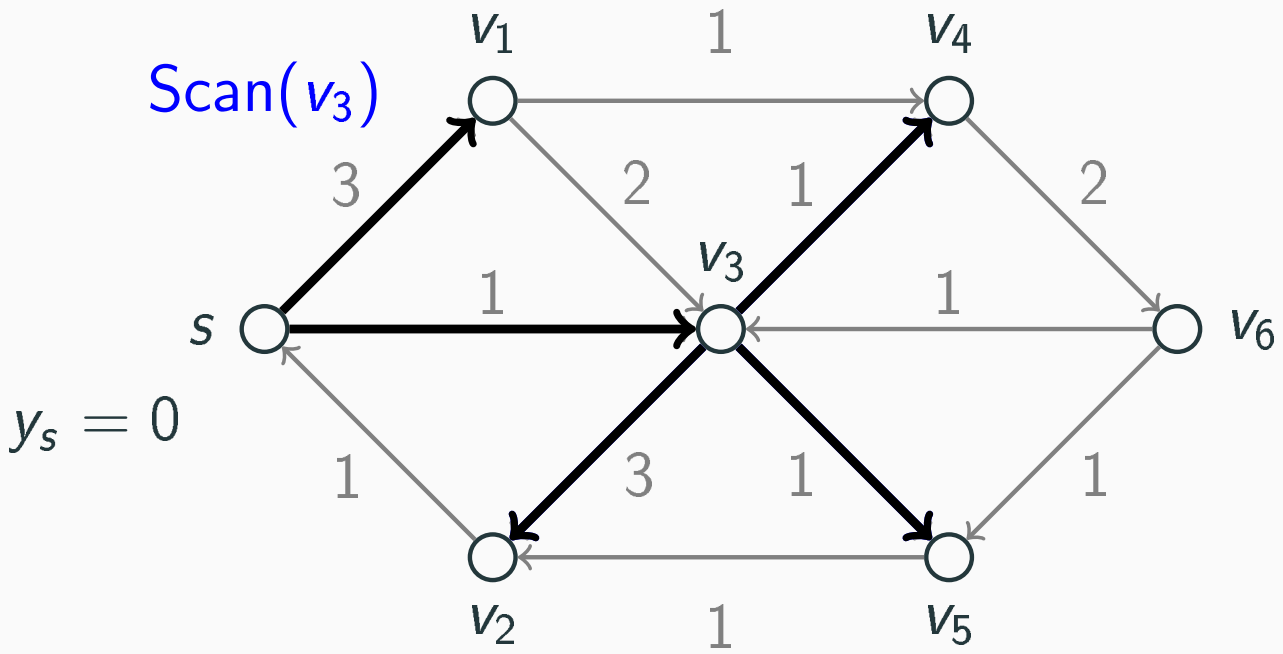
$y_{v3} = 1$



Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$ $\qquad$ $y_{v_4} = 2$

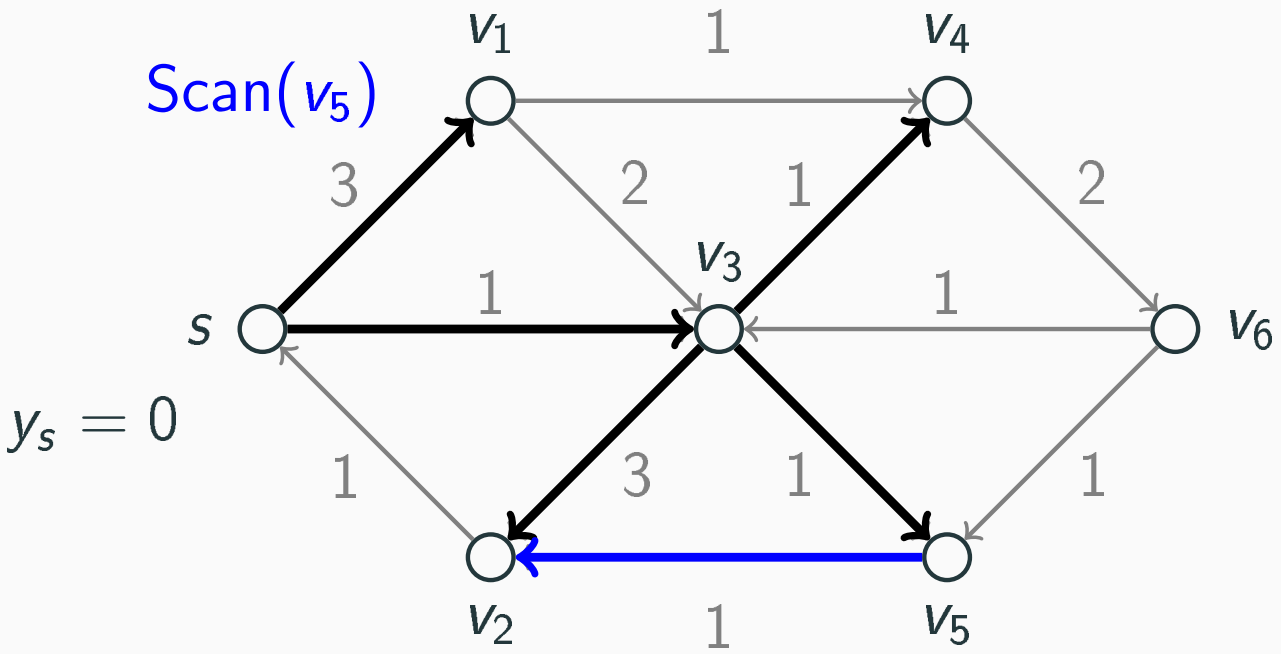


$y_{v_3} = 4$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v_3} = 1$ $\qquad$ $y_{v_4} = 2$

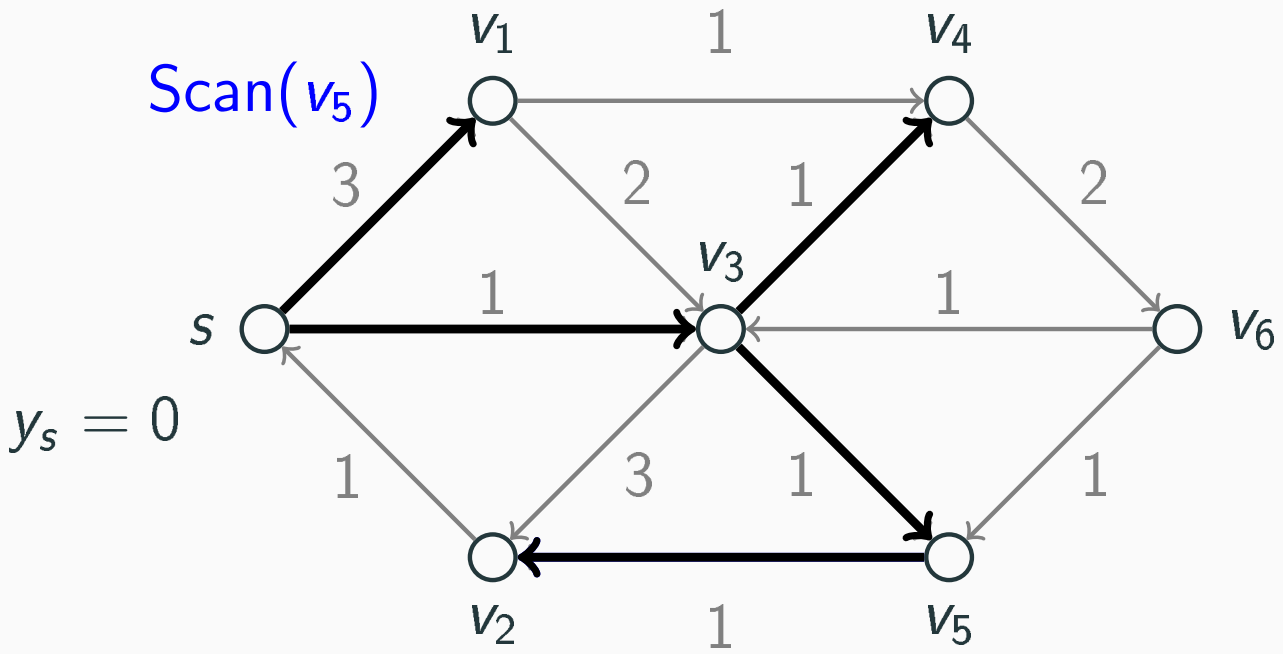


$y_{v_3} = 4$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$ $\qquad$ $y_{v_4} = 2$

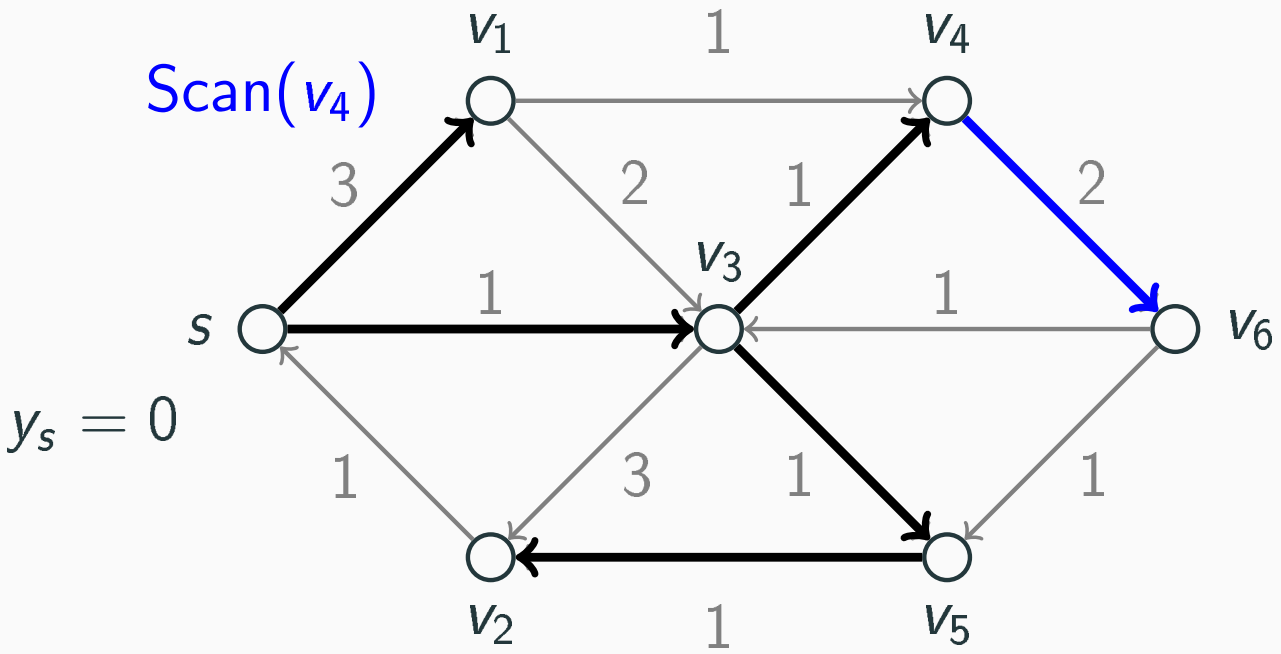


$y_{v_3} = 3$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$ $\qquad$ $y_{v_4} = 2$

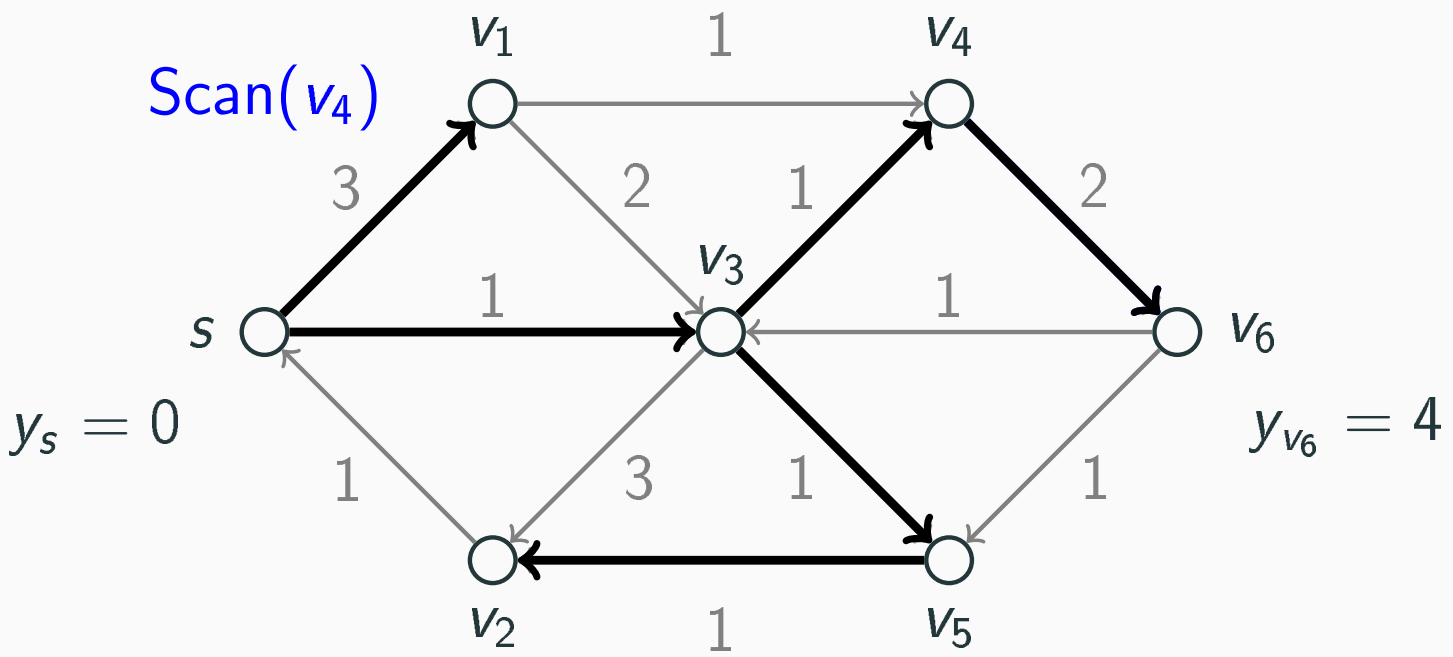


$y_{v_3} = 3$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$ $\qquad$ $y_{v_4} = 2$

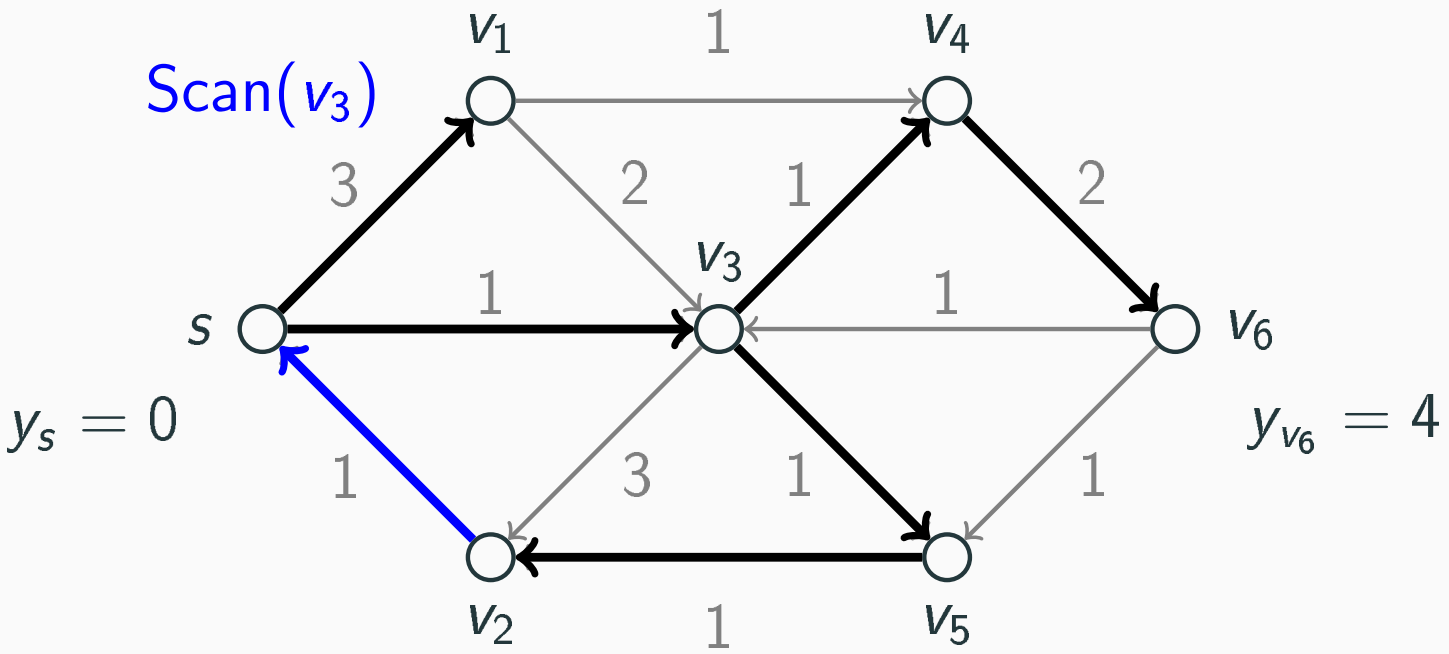


$y_{v_3} = 3$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$ $\qquad$ $y_{v_4} = 2$

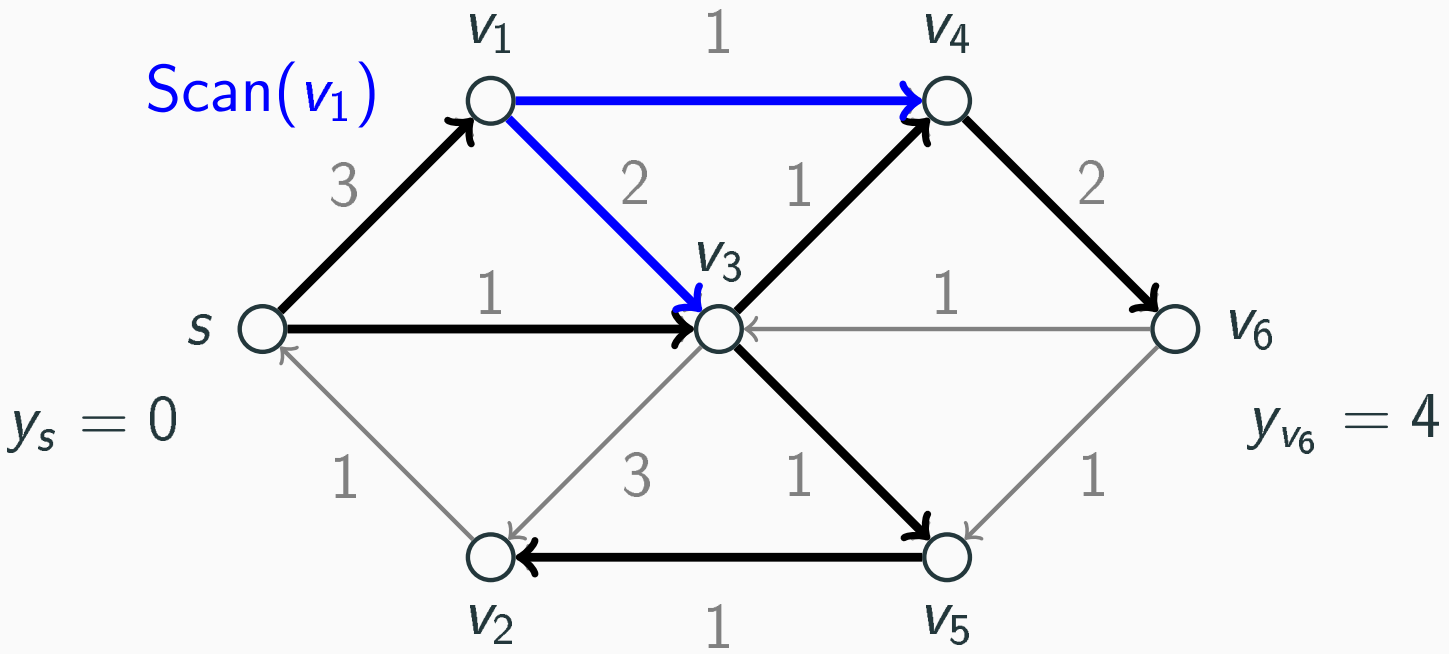


$y_{v_3} = 3$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$ $\qquad$ $y_{v_4} = 2$

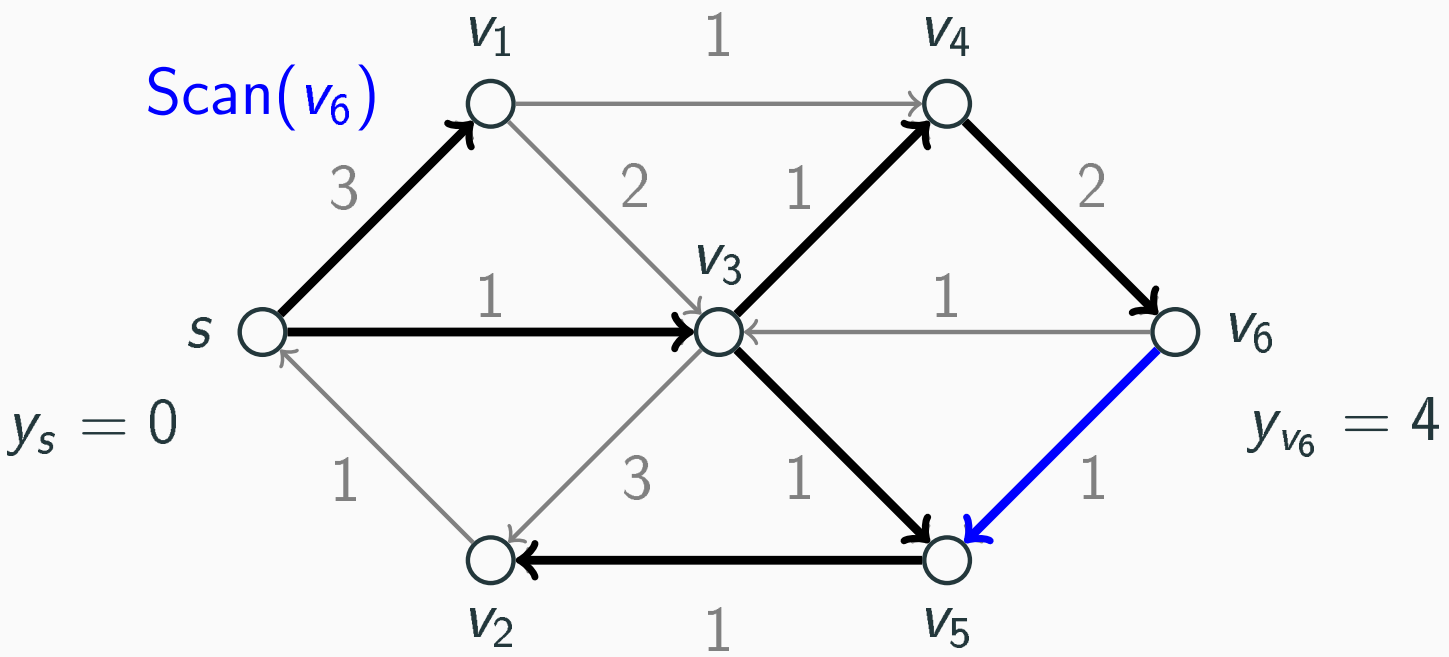


$y_{v_3} = 3$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$ $\qquad$ $y_{v_4} = 2$



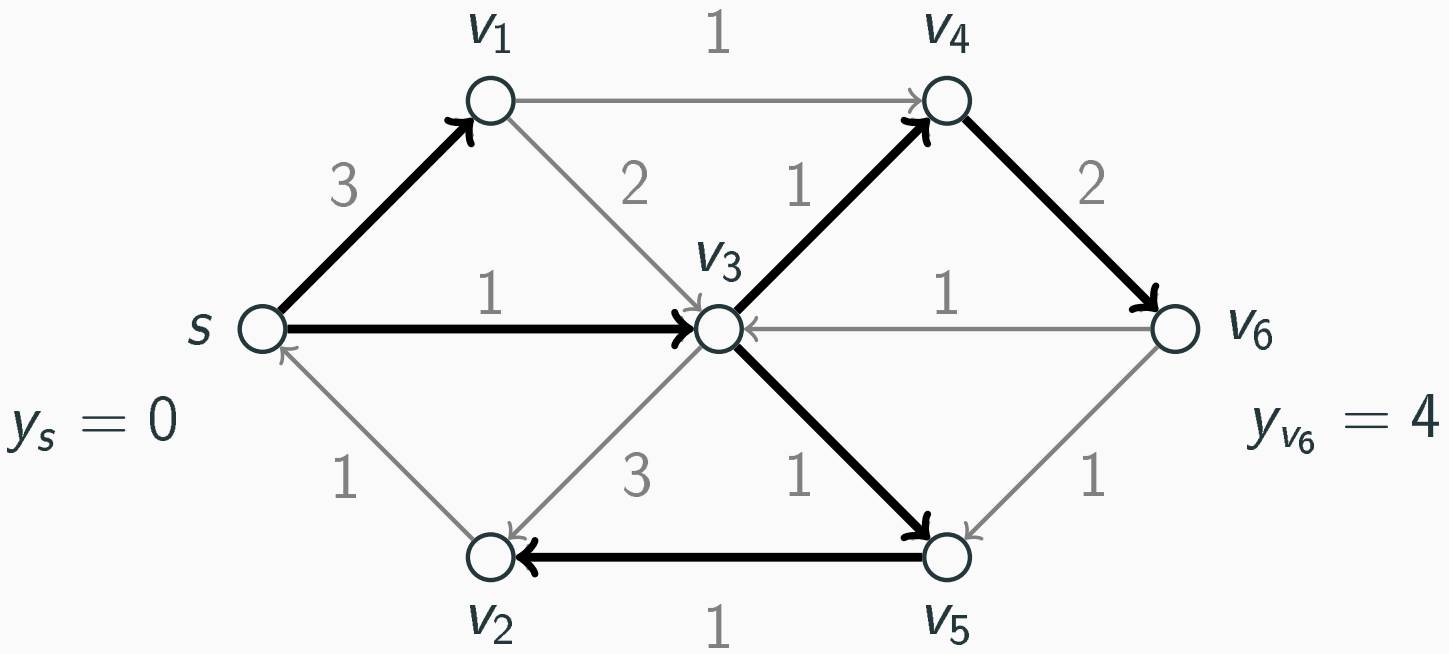
$y_{v_3} = 3$ $\qquad$ $y_{v_5} = 2$

Exécution de l'Algo de Dijkstra

$y_{v_1} = 3$

$y_{v3} = 1$

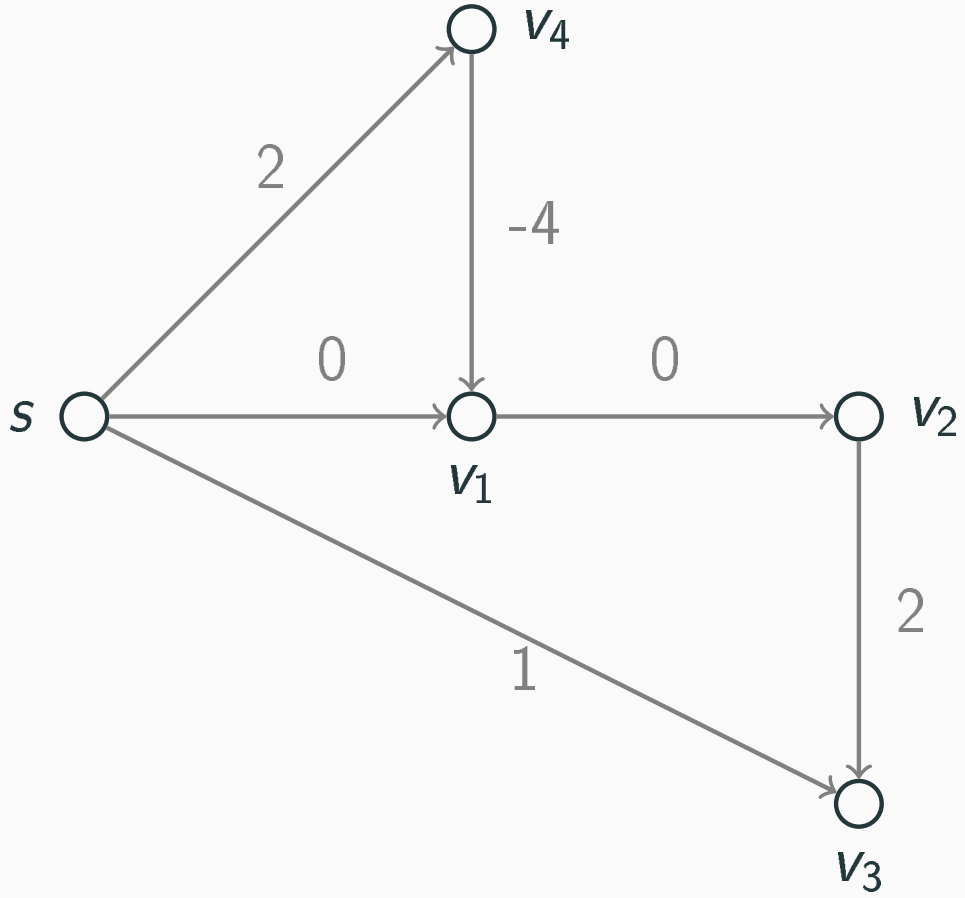
$y_{v_4} = 2$



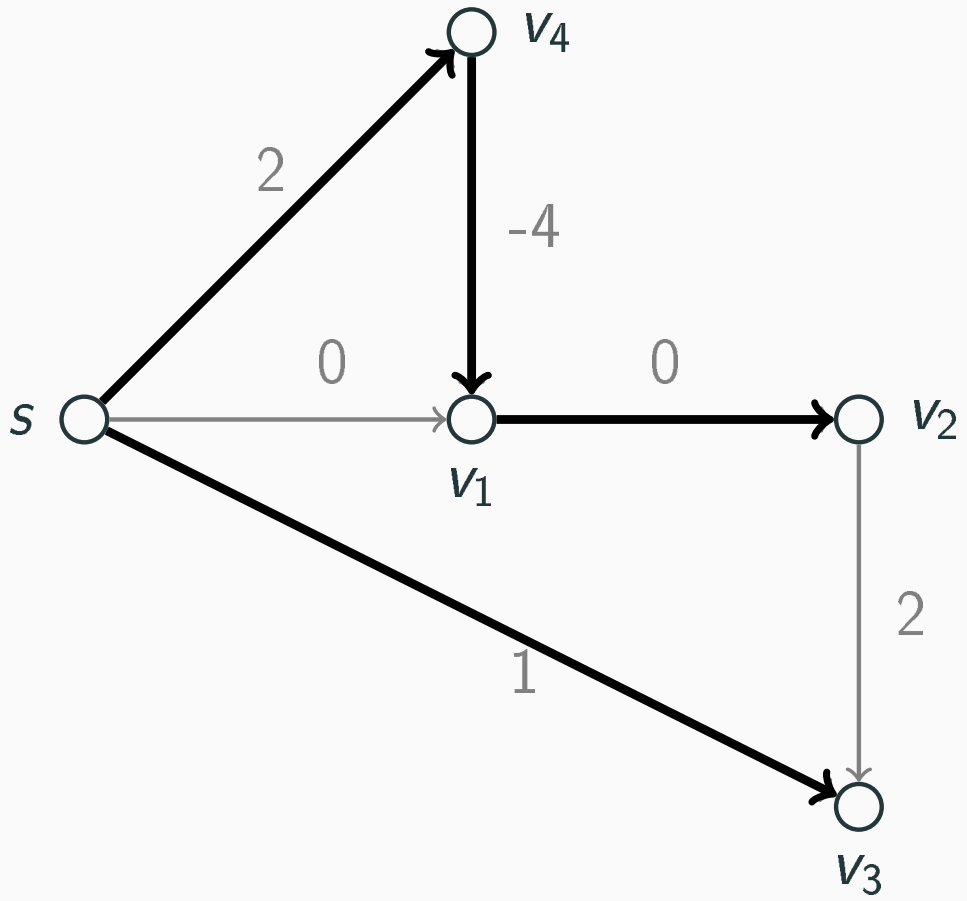
$y_{v_3} = 3$

$y_{v_5} = 2$

Algo de Dijkstra : Coûts Négatifs ? ?



Algo de Dijkstra : Coûts Négatifs ? ?



Dijkstra trouve la s-arborescence indiquée, qui n'est pas optimale.

Algo de Dijkstra : Justification

Lorsque l'algo choisit le sommet v, on a $y_v = \text{dist}(s, v)$.

Exercice

Mme Dupont sait d'expérience qu'elle ne doit pas garder une même voiture plus de 5 ans. Elle achète une voiture neuve, la garde un certain temps puis la revend pour en acheter une autre. Elle peut estimer le coût (= prix d'achat + coût d'utilisation − prix de vente) en fonction de l'année d'achat et de l'année de vente :

	2023	2024	2025	2026	2027
2022	1	2	4	7	10
2023		2	3	5	8
2024			2	3	6
2025				4	5
2026					4

Quelle est la meilleure politique de renouvellement ?

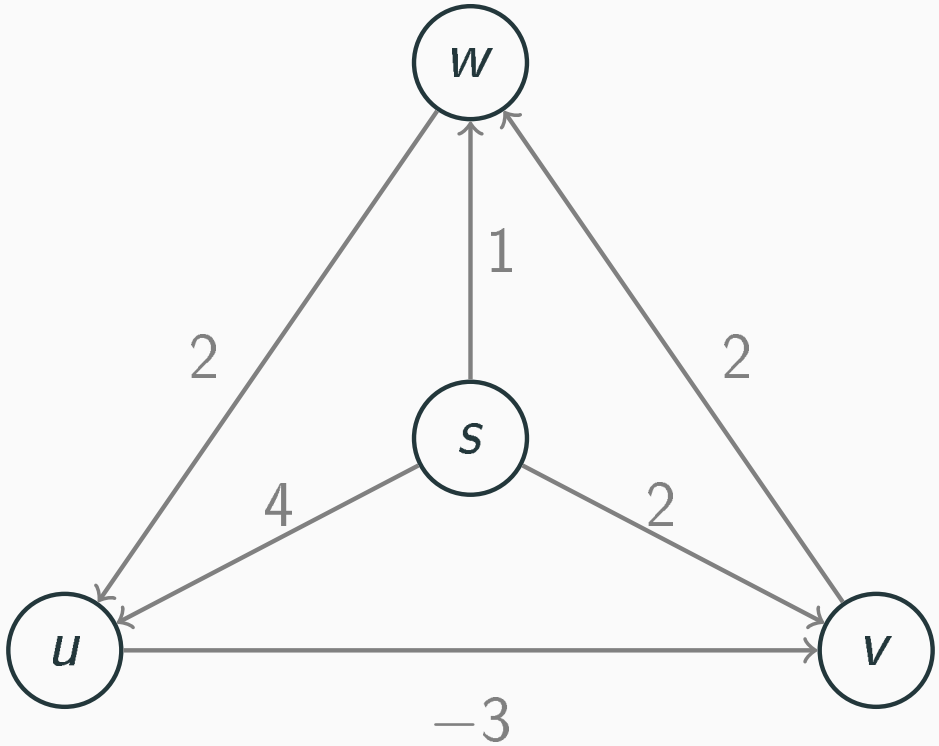
Réseaux sans Circuit Absorbant : Bellman-Ford

Algorithme 4 : Algo de Bellman-Ford

Entrée : Réseau $G = (V, A)$, $c : A \to \mathbb{R}$, sommet $s \in V$
Sortie : s-arborescence F

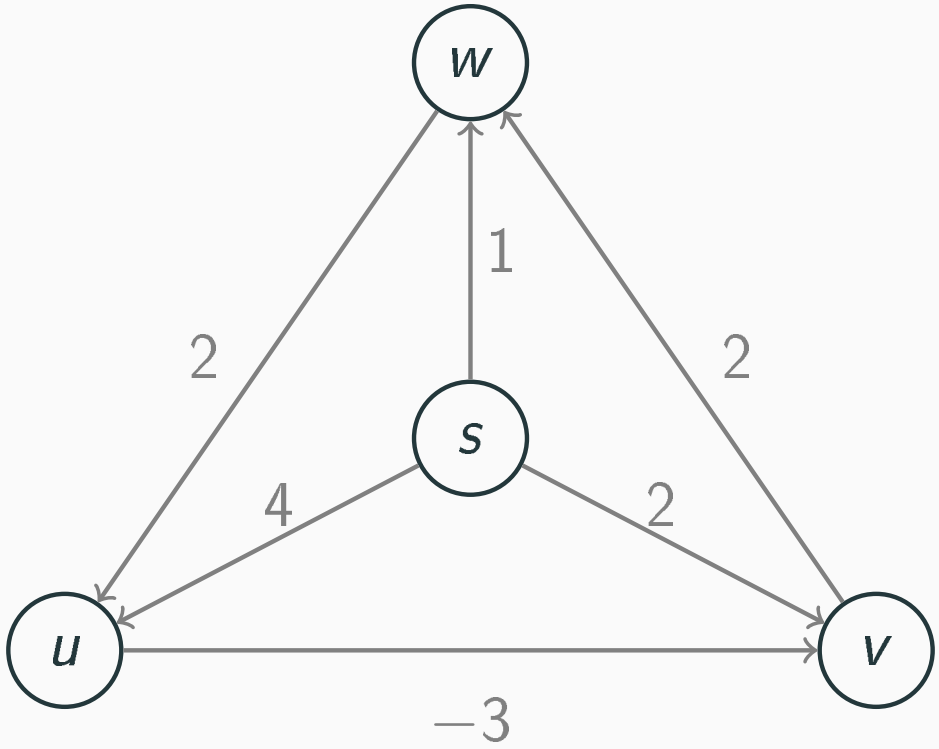
1 $y_s \leftarrow 0$ et $y_v \leftarrow \infty$ pour $v \in V - s$
2 $p_s \leftarrow 0$ et $p_v \leftarrow -1$ pour $v \in V - s$
3 **pour** $0 \leq i \leq |V|$ **faire**
4 **pour** $v \in V$ **faire**
5 Scan(v)
6 **si** Scan($\cdot$) *n'a pas changé y dans cette itération* **alors**
7 $A_F \leftarrow \{p_v v \mid v \in V - s\}$
8 **return** $F \leftarrow (V, A_F)$
9 Afficher "G, c contient un circuit absorbant"
10 **fin**

Bellman-Ford : Exécution de l'Algorithme



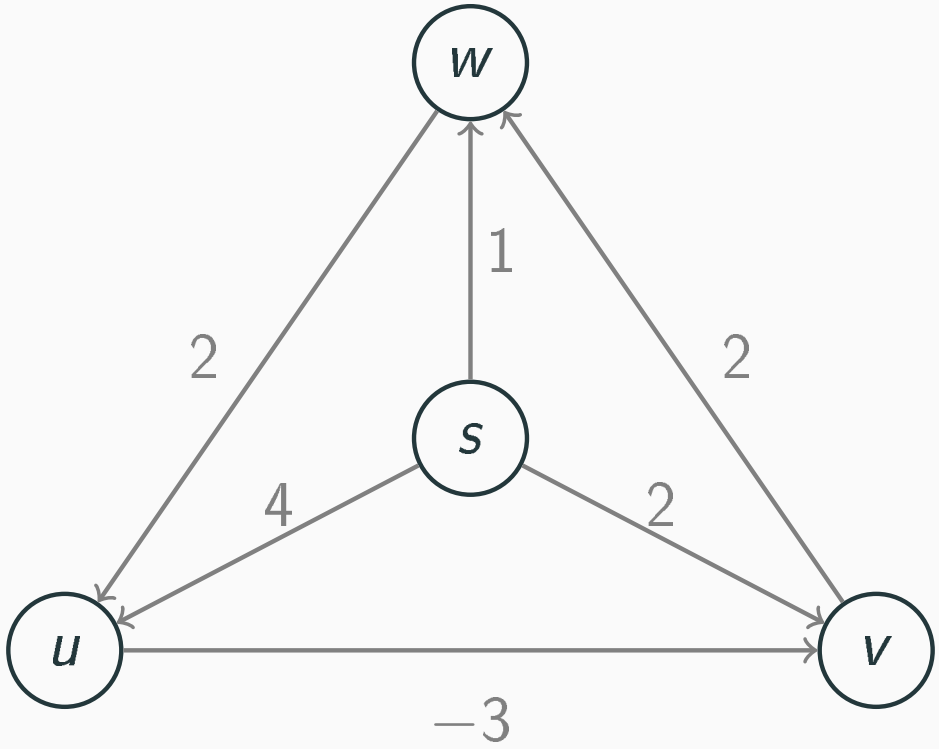
Scan(·)	y_s	y_u	y_v	y_w

Bellman-Ford : Exécution de l'Algorithme



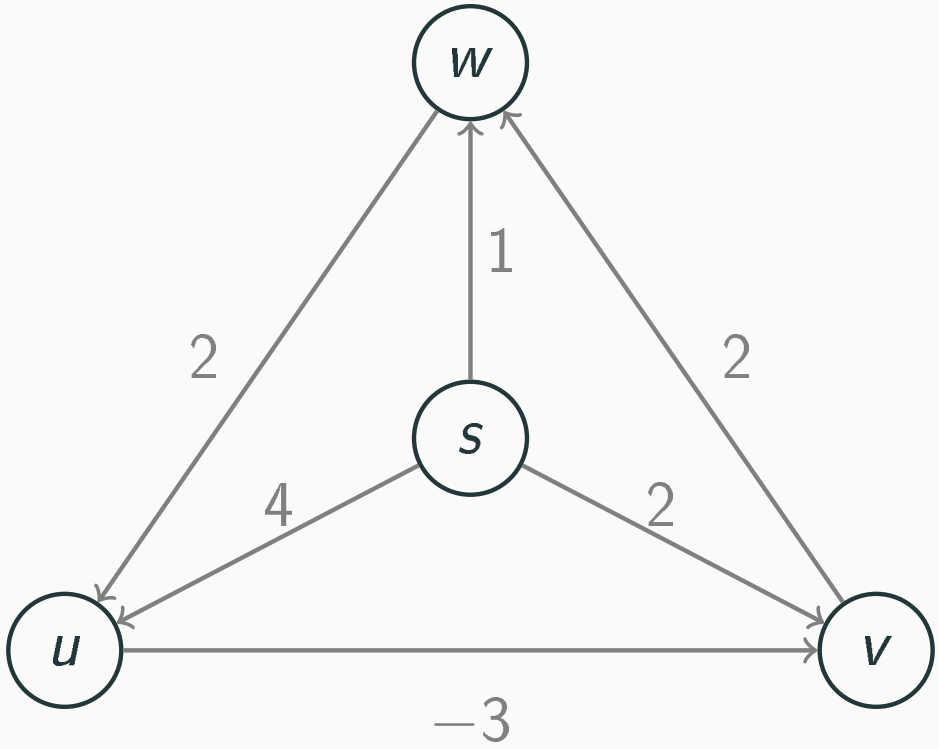
Scan(·)	y_s	y_u	y_v	y_w
s	0	4	2	1

Bellman-Ford : Exécution de l'Algorithme



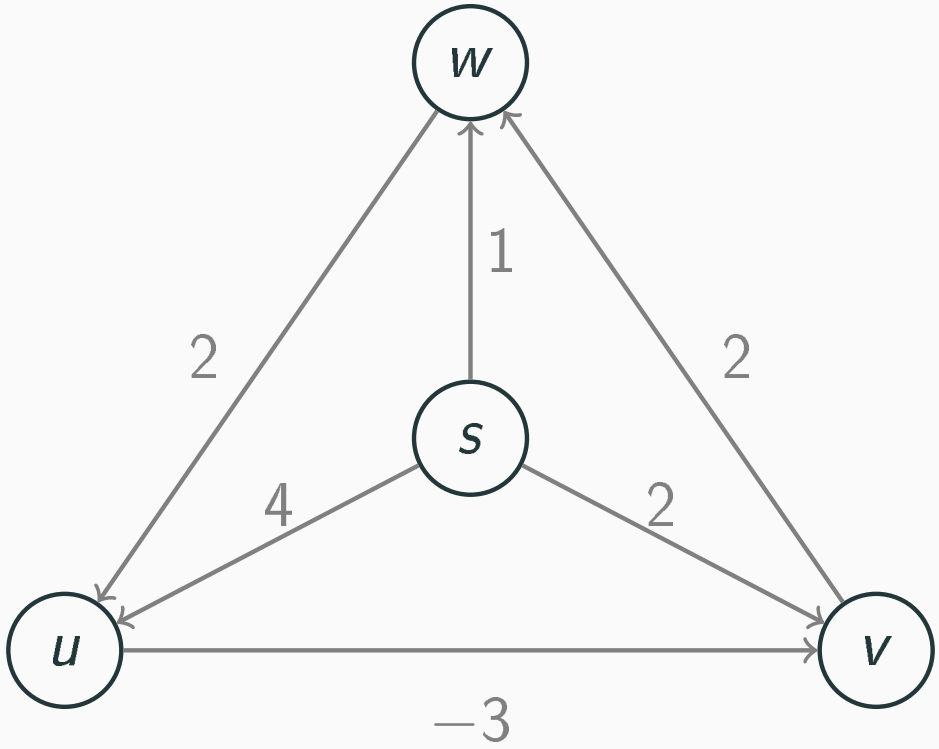
Scan(·)	y_s	y_u	y_v	y_w
s	0	4	2	1
u	0	4	1	1

Bellman-Ford : Exécution de l'Algorithme



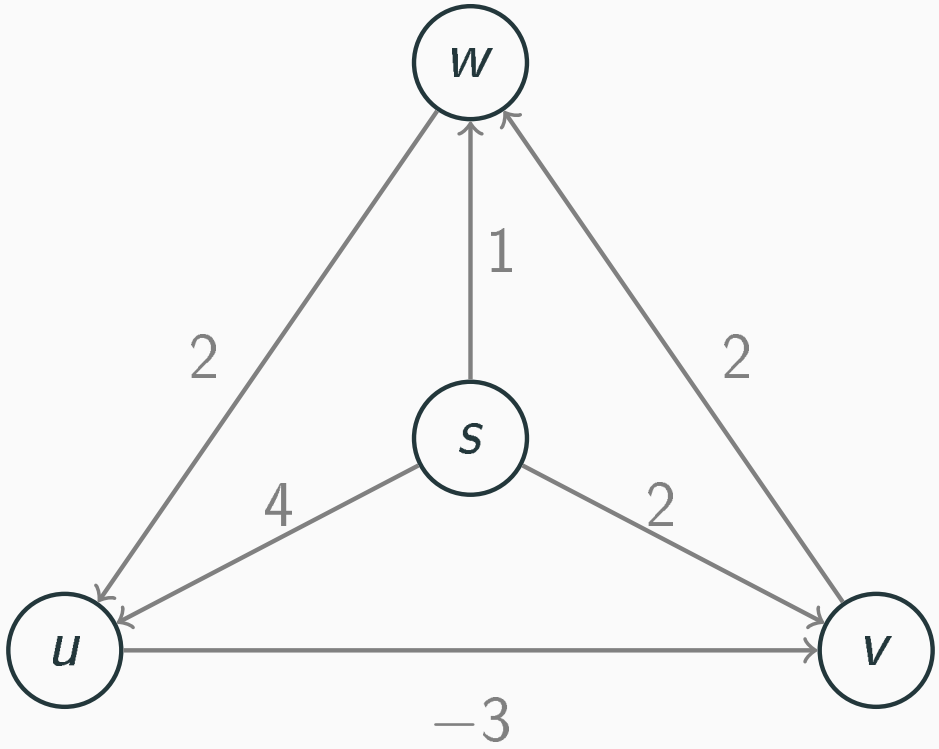
Scan(·)	y_s	y_u	y_v	y_w
s	0	4	2	1
u	0	4	1	1
v	0	4	1	1

Bellman-Ford : Exécution de l'Algorithme



Scan(·)	y_s	y_u	y_v	y_w
s	0	4	2	1
u	0	4	1	1
v	0	4	1	1
w	0	3	1	1

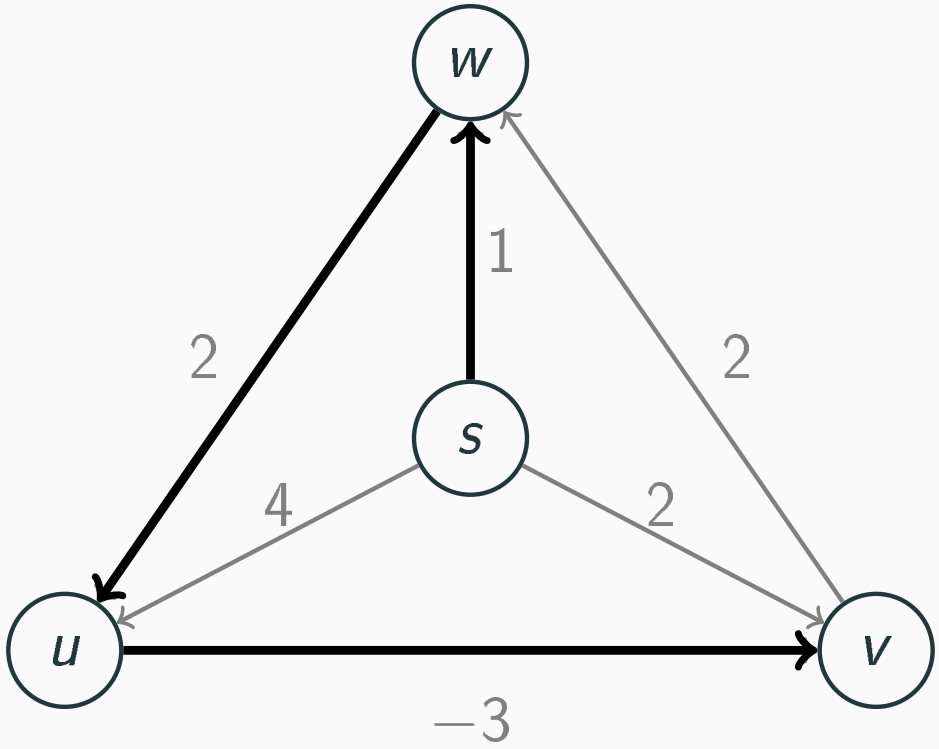
Bellman-Ford : Exécution de l'Algorithme



Scan(·)	y_s	y_u	y_v	y_w
s	0	4	2	1
u	0	4	1	1
v	0	4	1	1
w	0	3	1	1

Scan(·)	y_s	y_u	y_v	y_w
s	0	3	1	1
u	0	3	0	1
v	0	3	0	1
w	0	3	0	1

Bellman-Ford : Exécution de l'Algorithme



Scan(·)	y_s	y_u	y_v	y_w
s	0	4	2	1
u	0	4	1	1
v	0	4	1	1
w	0	3	1	1

Scan(·)	y_s	y_u	y_v	y_w
s	0	3	1	1
u	0	3	0	1
v	0	3	0	1
w	0	3	0	1

Bellman-Ford : Justification

Bellman-Ford trouve un plus court s-v chemin pour tout v où indique correctement qu'un circuit absorbant existe.

Bellman-Ford : Programmation Dynamique

- Sous problèmes : calculer $w_k(v)$, le coût minimum d'un s-v chemin ayant au plus k arcs

Bellman-Ford : Programmation Dynamique

- Sous problèmes : calculer $w_k(v)$, le coût minimum d'un s-v chemin ayant au plus k arcs
- Au début, on a $w_0(s) = 0$ et $w_0(v) = \infty$ pour tout $v \in V - s$.

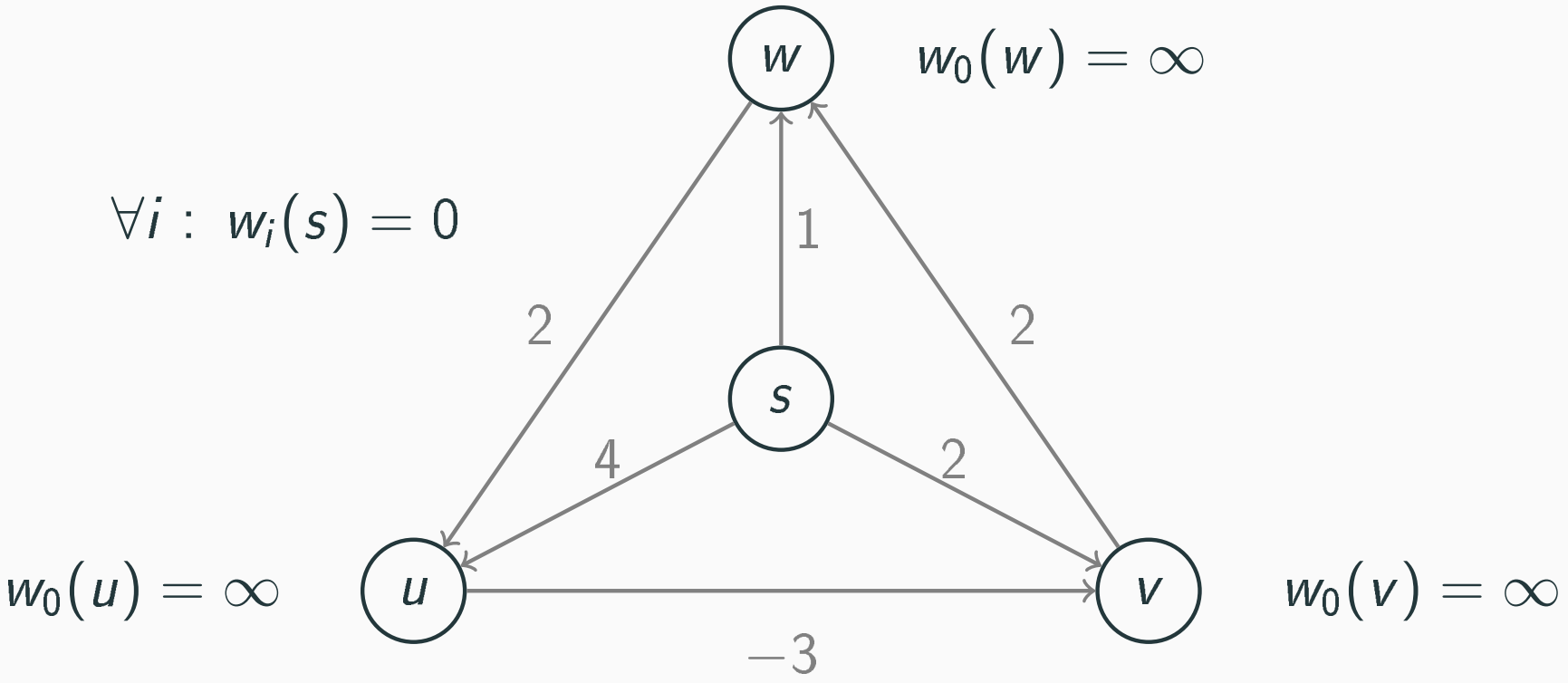
Bellman-Ford : Programmation Dynamique

- Sous problèmes : calculer $w_k(v)$, le coût minimum d'un s-v chemin ayant au plus k arcs
- Au début, on a $w_0(s) = 0$ et $w_0(v) = \infty$ pour tout $v \in V - s$.
- Après l'itération i, l'algo de Bellman a calculé pour $v \in V$:

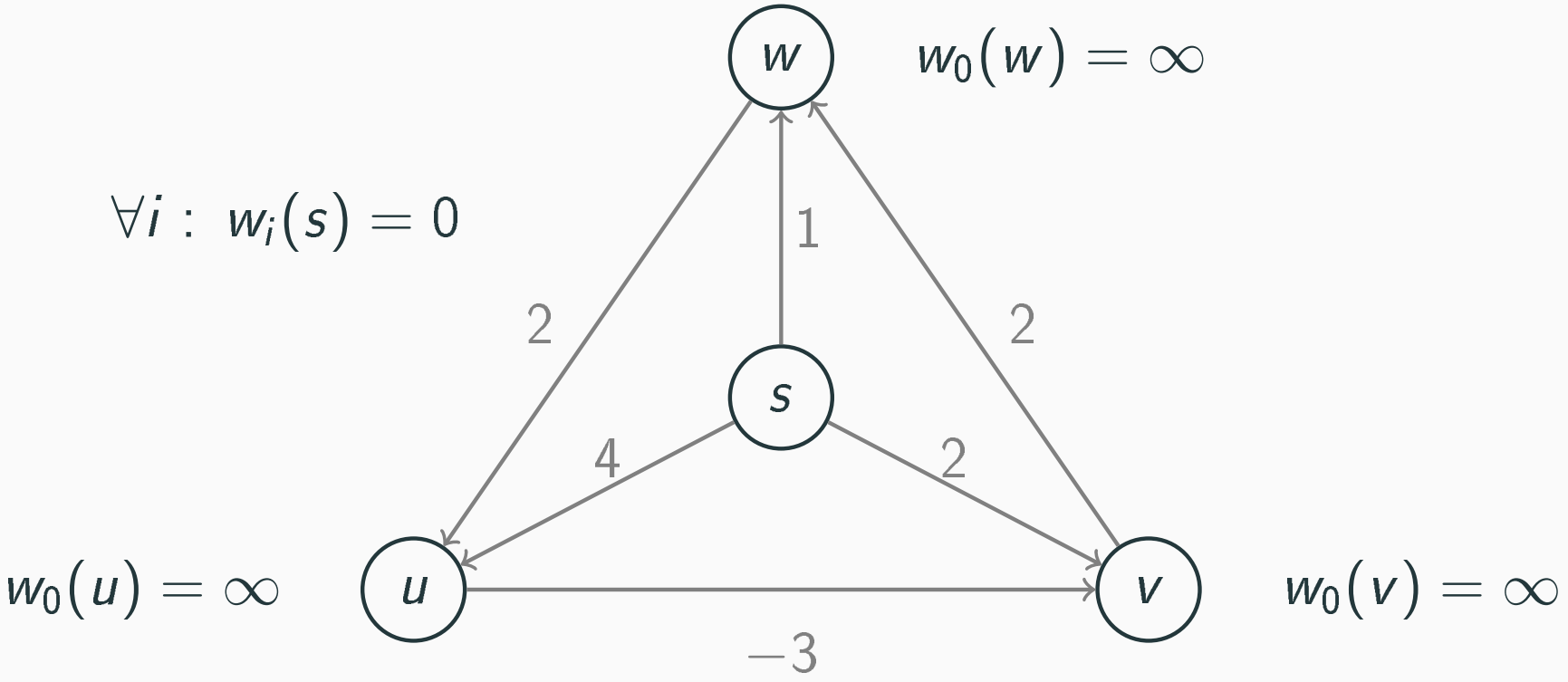
$$w_{i+1}(v) = \min_{uv \in A} \{w_i(v), w_i(u) + c(uv)\}$$

- À la fin : $w_{|V|-1}(v) = w_{|V|}(v) = \text{dist}(s, v)$ pour tout $v \in V$ si G, c est sans circuit absorbant.

Bellman-Ford : Programmation Dynamique

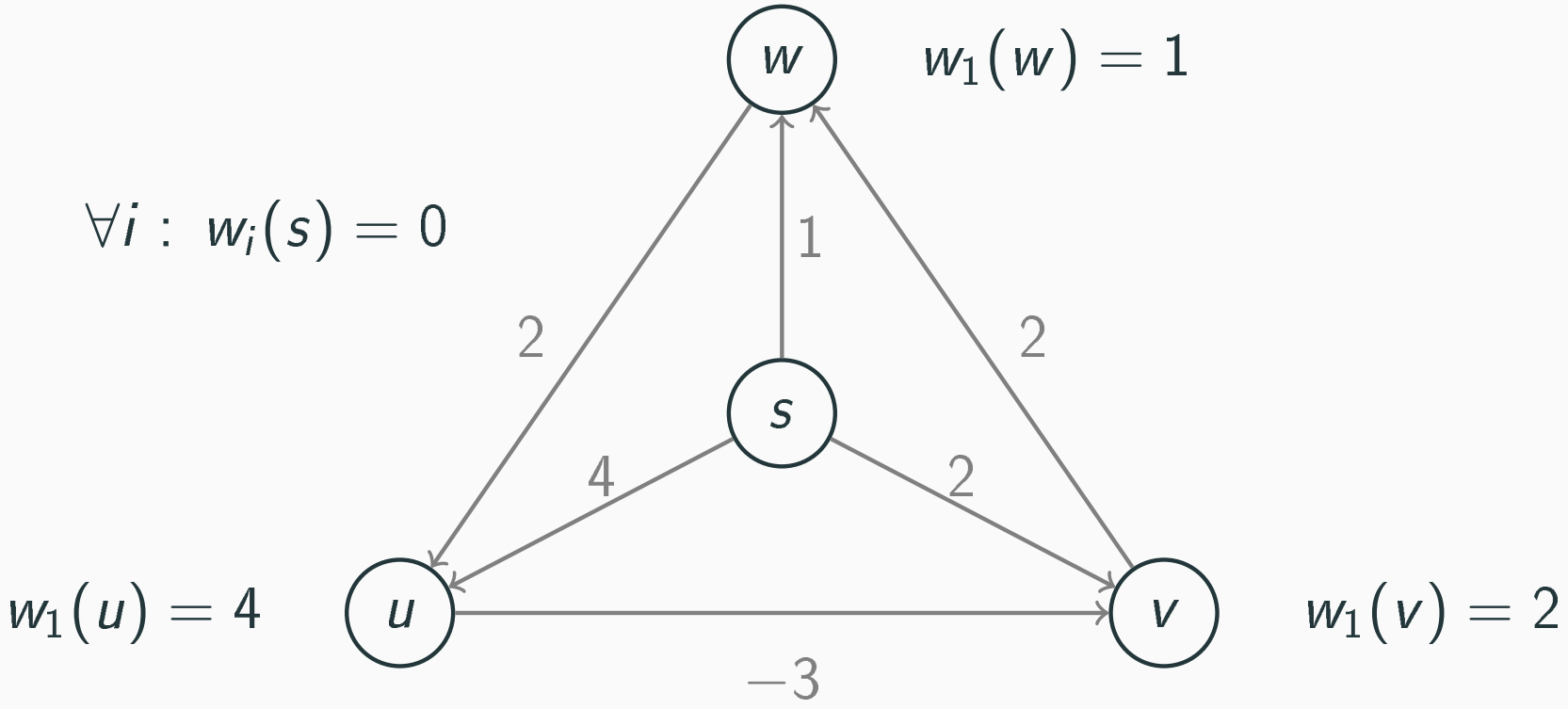


Bellman-Ford : Programmation Dynamique



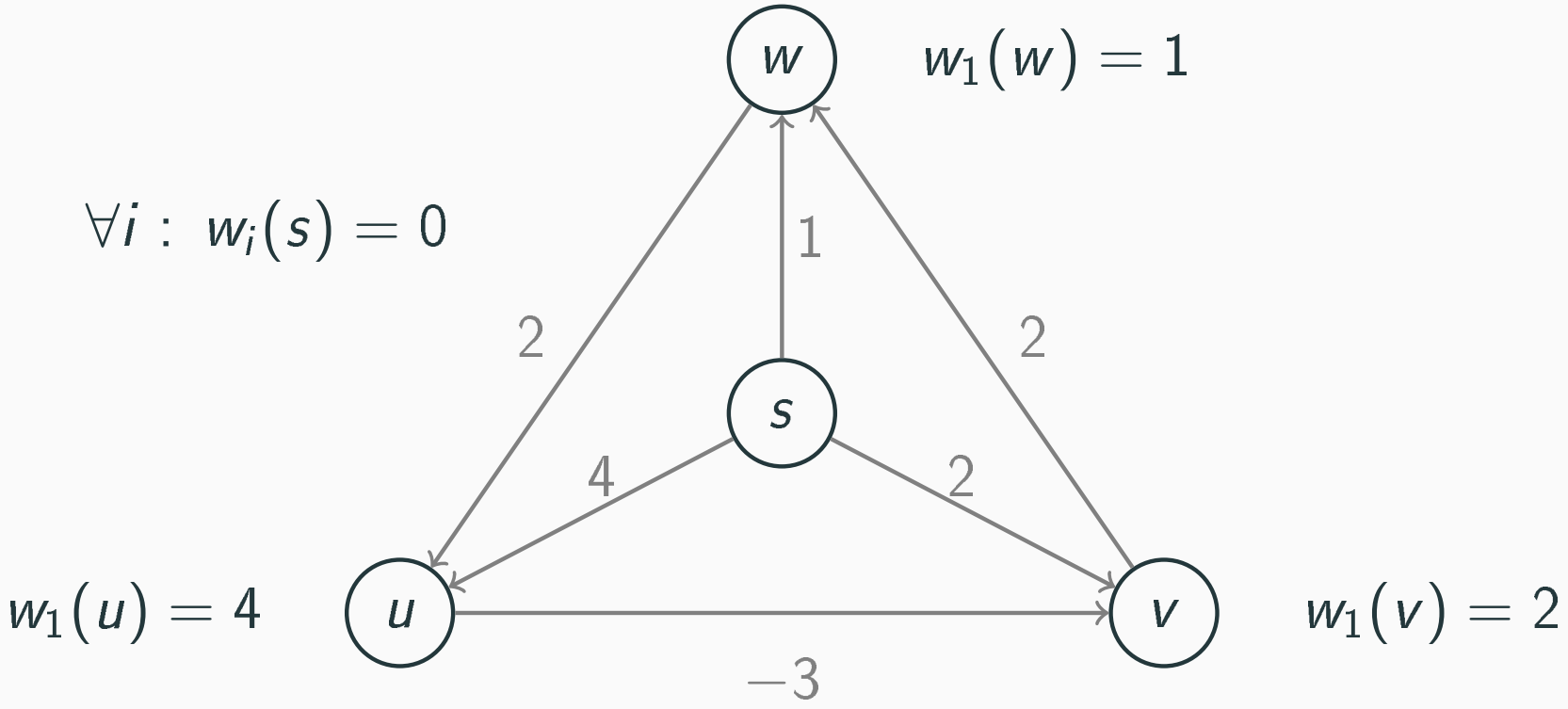
$$w_1(u) = \min\{w_0(u), w_0(s) + c(su), w_0(w) + c(wu)\} = 4$$
$$w_1(v) = \min\{w_0(v), w_0(s) + c(sv), w_0(u) + c(uv)\} = 2$$
$$w_1(w) = \min\{w_0(w), w_0(s) + c(sw), w_0(v) + c(vw)\} = 1$$

Bellman-Ford : Programmation Dynamique



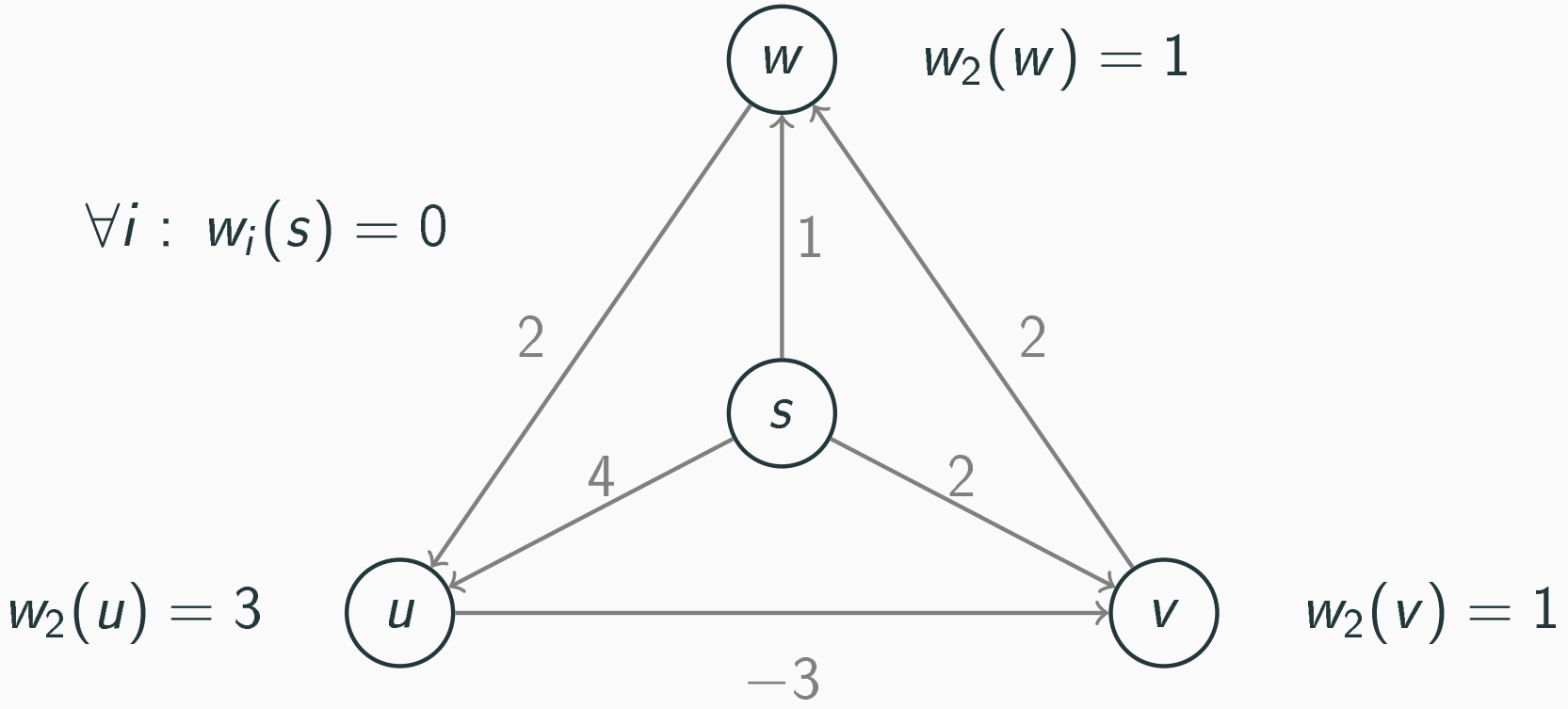
$$w_1(u) = \min\{w_0(u), w_0(s) + c(su), w_0(w) + c(wu)\} = 4$$
$$w_1(v) = \min\{w_0(v), w_0(s) + c(sv), w_0(u) + c(uv)\} = 2$$
$$w_1(w) = \min\{w_0(w), w_0(s) + c(sw), w_0(v) + c(vw)\} = 1$$

Bellman-Ford : Programmation Dynamique



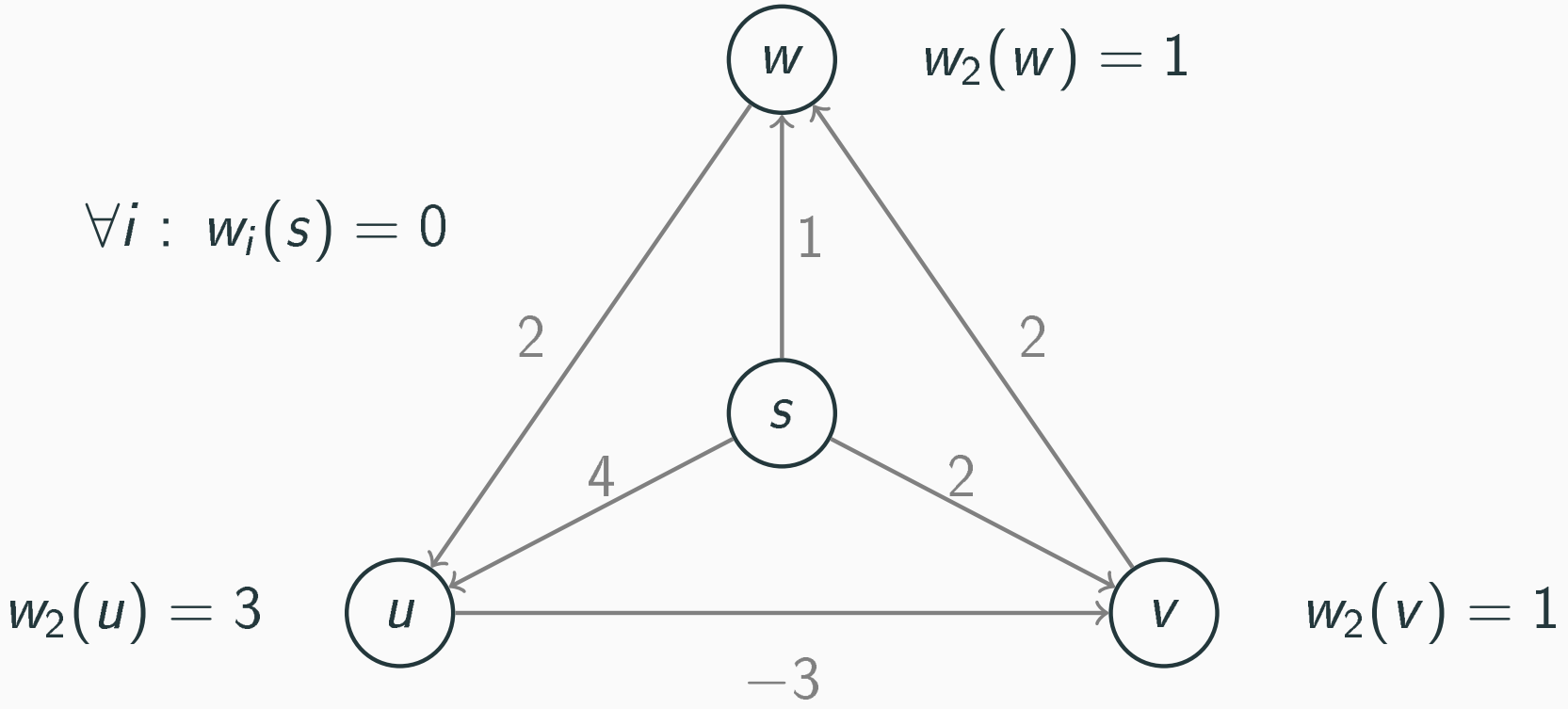
$$w_2(u) = \min\{w_1(u), w_1(s) + c(su), w_1(w) + c(wu)\} = 3$$
$$w_2(v) = \min\{w_1(v), w_1(s) + c(sv), w_1(u) + c(uv)\} = 1$$
$$w_2(w) = \min\{w_1(w), w_1(s) + c(sw), w_1(v) + c(vw)\} = 1$$

Bellman-Ford : Programmation Dynamique



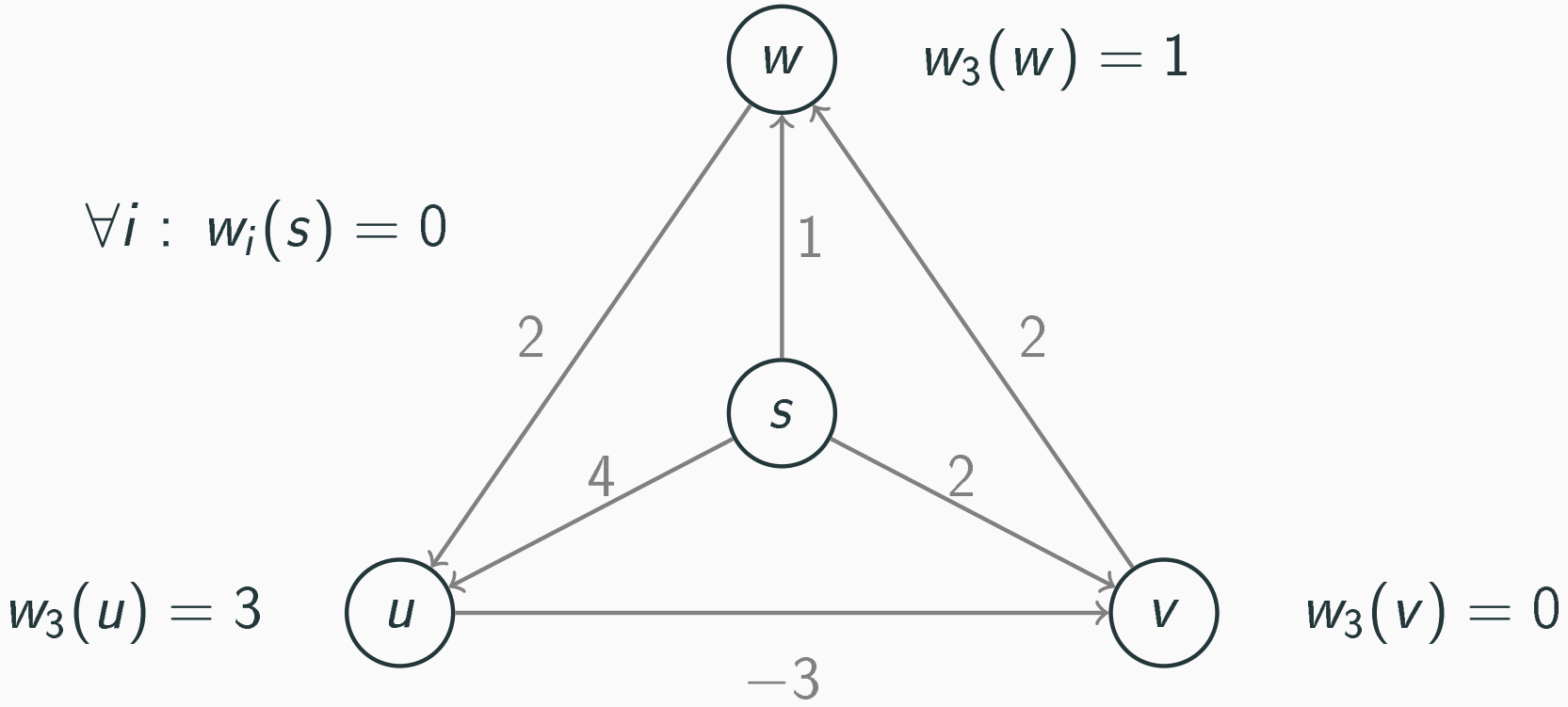
$$w_2(u) = \min\{w_1(u), w_1(s) + c(su), w_1(w) + c(wu)\} = 3$$
$$w_2(v) = \min\{w_1(v), w_1(s) + c(sv), w_1(u) + c(uv)\} = 1$$
$$w_2(w) = \min\{w_1(w), w_1(s) + c(sw), w_1(v) + c(vw)\} = 1$$

Bellman-Ford : Programmation Dynamique



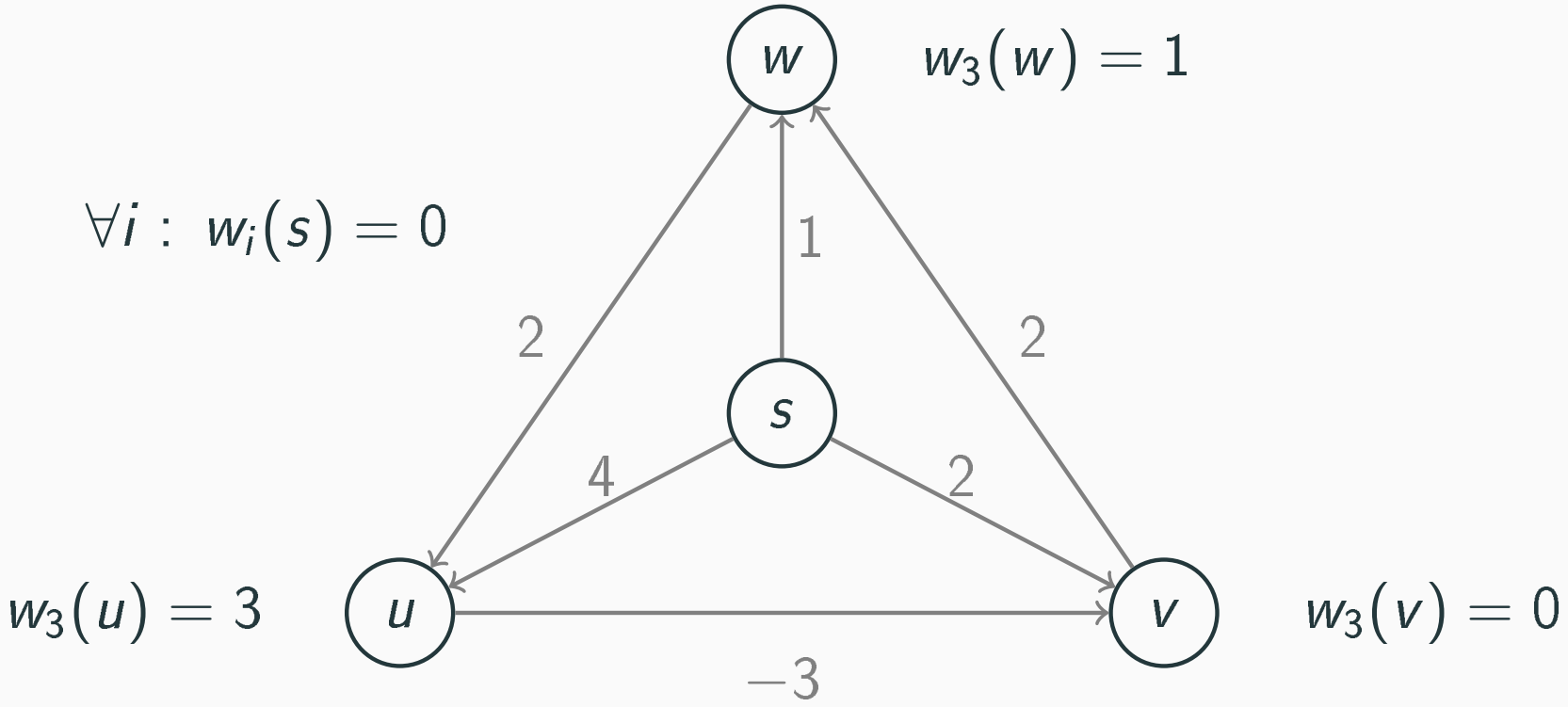
$$w_3(u) = \min\{w_2(u), w_2(s) + c(su), w_2(w) + c(wu)\} = 3$$
$$w_3(v) = \min\{w_2(v), w_2(s) + c(sv), w_2(u) + c(uv)\} = 0$$
$$w_3(w) = \min\{w_2(w), w_2(s) + c(sw), w_2(v) + c(vw)\} = 1$$

Bellman-Ford : Programmation Dynamique



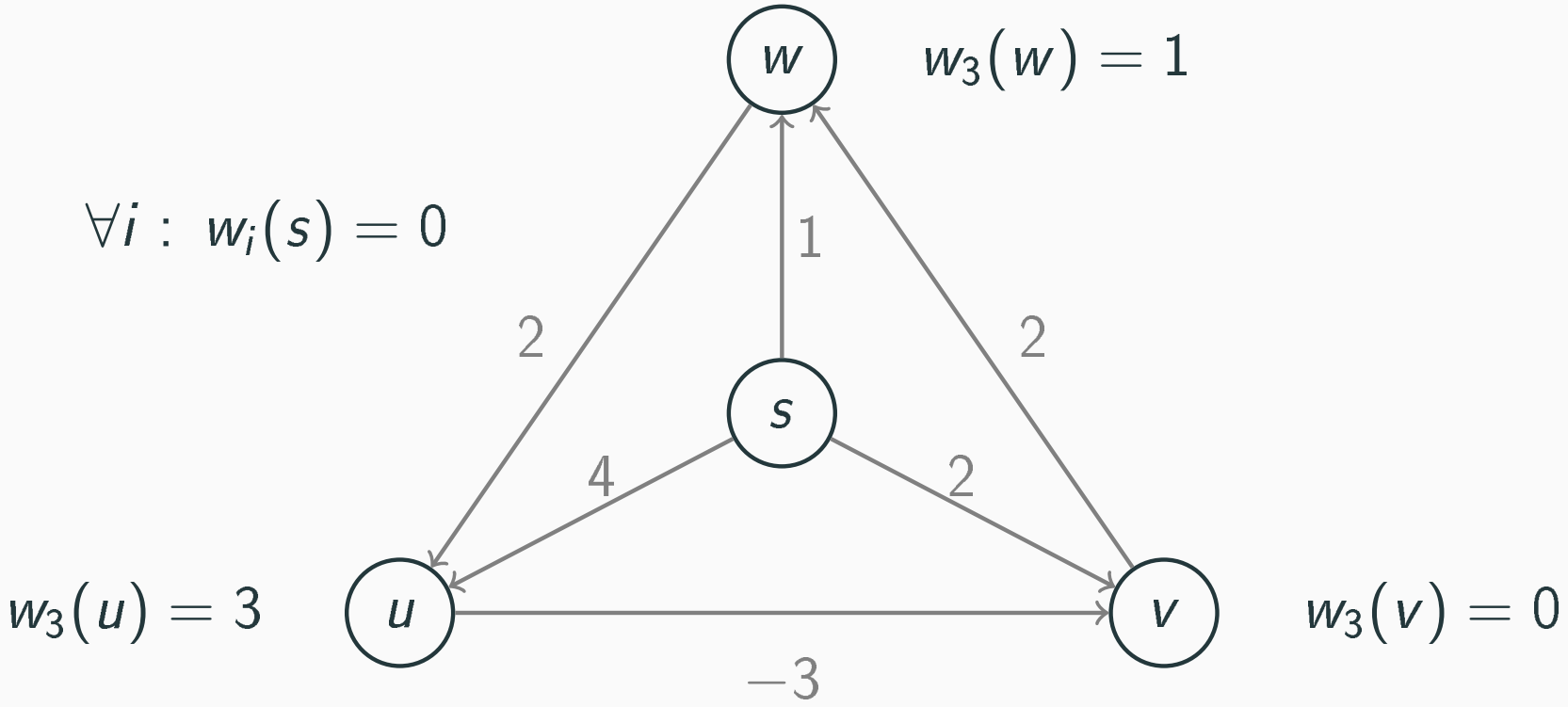
$$w_3(u) = \min\{w_2(u), w_2(s) + c(su), w_2(w) + c(wu)\} = 3$$
$$w_3(v) = \min\{w_2(v), w_2(s) + c(sv), w_2(u) + c(uv)\} = 0$$
$$w_3(w) = \min\{w_2(w), w_2(s) + c(sw), w_2(v) + c(vw)\} = 1$$

Bellman-Ford : Programmation Dynamique



$$w_4(u) = \min\{w_3(u), w_3(s) + c(su), w_3(w) + c(wu)\} = w_3(u)$$
$$w_4(v) = \min\{w_3(v), w_3(s) + c(sv), w_3(u) + c(uv)\} = w_3(v)$$
$$w_4(w) = \min\{w_3(w), w_3(s) + c(sw), w_3(v) + c(vw)\} = w_3(w)$$

Bellman-Ford : Programmation Dynamique



$$w_4(u) = \min\{w_3(u), w_3(s) + c(su), w_3(w) + c(wu)\} = w_3(u)$$
$$w_4(v) = \min\{w_3(v), w_3(s) + c(sv), w_3(u) + c(uv)\} = w_3(v)$$
$$w_4(w) = \min\{w_3(w), w_3(s) + c(sw), w_3(v) + c(vw)\} = w_3(w)$$