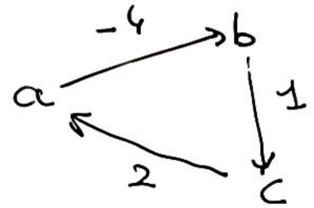
Condition d'existence du plus court chemin :

* il existe un chemin de s à t
* il n'y a pas de circuit absorbant.

<u>Définition</u> : Circuit absorbant

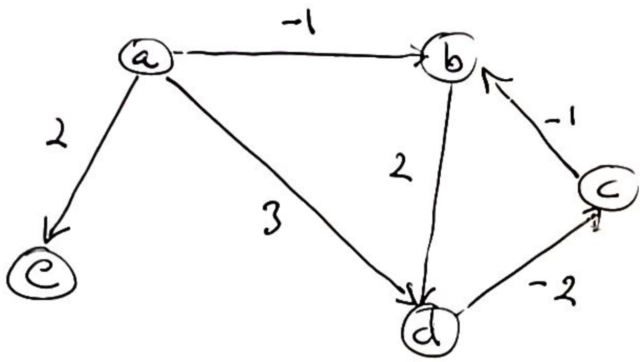
Un circuit absorbant un un circuit de longueur strictement négative.



$$\boxed{\sum_{u \in U} l(u) < 0}$$

Dans cette exemple $\sum_{u \in U} l(u) = -4 + 1 + 2 = -1 < 0$

Considérons $G = (X, U)$



On souhaite determiner le plus court chemin de a vers c par exemple.

adc $= 1$

adcbdc $= 0$

adcbdcbdc $= -1$

$\vdots$

adcbdc....bdc $= -\infty$

Ainsi l'idée d'avoir un plus court chemin n'a aucun sens puisque l'on dispose d'un circuit absorbant dans le graphe.

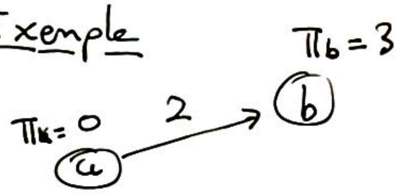
Définition Potentiel

Soit $G = (X, U)$ un graphe

On note $\pi : X \to \mathbb{R}$. π est un potentiel

si $\forall\, ij \in U \quad \pi(j) - \pi(i) \leq \ell_{ij} \iff \pi_j \leq \pi_i + \ell_{ij}$

Exemple



π n'est pas un potentiel

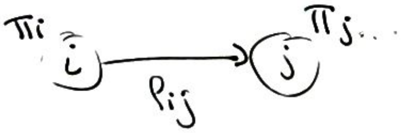
π n'est pas un potentiel $\iff \exists\, ij \in U \quad \pi(j) - \pi(i) > \ell_{ij}$

$\iff \pi_j > \pi_i + \ell_{ij}$

Algorithme de Ford

⚠ Cas où on cherche à minimiser la longueur.

L'idée est de parcourir toute les arêtes et de se demander si le potentiel du sommet i + la longueur de i à j est plus petite que le potentiel de j



$\pi_j > \pi_i + \ell_{ij}$?

Si la condition est verifiée, on change le potentiel de j à $\pi_i + \ell_{ij}$ et i devient le prédécesseur de j.

On répète le processus $n-1$ fois où n est le nombre de sommet ou bien on s'arrête avant quand la condition n'est plus remplie : $\forall\, ij \in U \quad \pi_j \leq \pi_i + \ell_{ij}$

ce qui revient à dire que Π est un potentiel.

Algo : On cherche le + court chemin à partir de s

initialisation $\Pi_i = \begin{cases} 0 & \text{si } i = s \\ l_{si} & \text{si } i \text{ successeur de } s \\ +\infty & \text{sinon} \end{cases}$

$pred_i = \begin{cases} s & \text{si } i = s \text{ ou } i \text{ successeur de } s \\ \emptyset & \text{sinon} \end{cases}$

$k \leftarrow 1$

Tant que $(k < n)$ et (Π n'est pas un potentiel)

Pour tout $ij \in U$ faire

Si $\Pi_j > \Pi_i + l_{ij}$

$\Pi_j \leftarrow \Pi_i + l_{ij}$

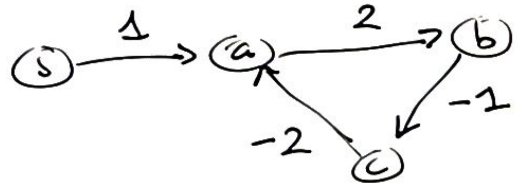
$pred_j \leftarrow i$

Fin Si

Fin Pour

Fin Tant Que

<u>Rq</u> Si à la fin de l'algorithme $k = n-1$ et Π n'est pas un potentiel ($\Leftrightarrow \exists\, ij \in U$ tq $\Pi_j > \Pi_i + l_{ij}$) alors le graphe contient un circuit absorbant. La solution de l'algorithme n'est donc pas bonne, on ne peut pas trouver de plus court chemin en partant de s.



Plus court chemin de s à b :

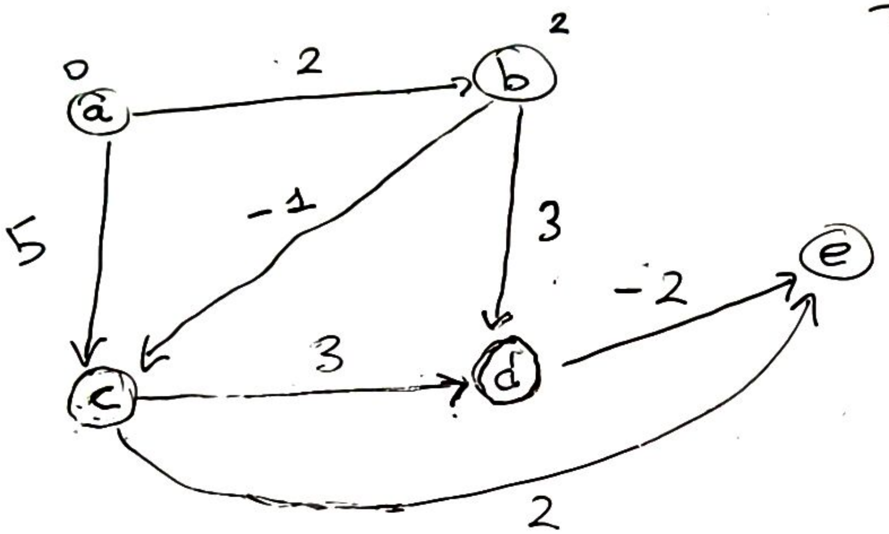
sab : 3

sabcab : 2

sabcabcab...cab : $-\infty$

Exemple

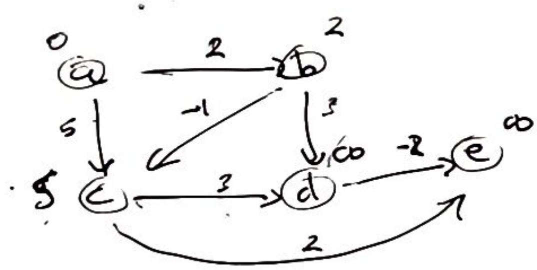
$$\pi_j > \pi_i + \ell_{ij} \ ?$$



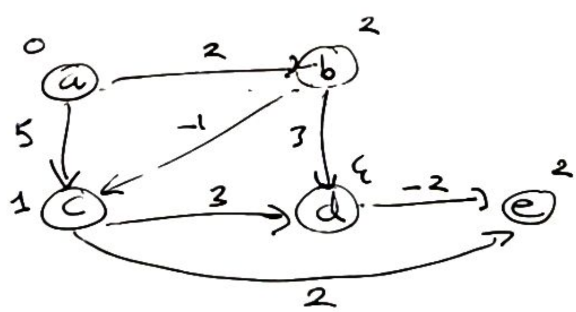
$U = \{ab, bc, ac, bd, cd, ce, de\}$

	a	b	c	d	e
π^0	0	2	5	$+\infty$	$+\infty$
pred	a	a	a	$\emptyset$	$\emptyset$

$k = 1$



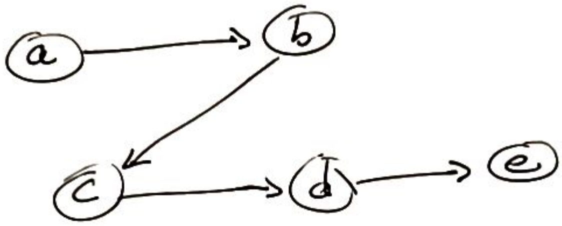
	a	b	c	d	e
π^1	0	2	1	4	2
pred	a	a	b	c	d



est ce que $\pi_j \leq \pi_i + \ell_{ij} \ \forall i j$?

oui ! on sort de l'algo

graphe du plus court chemin :



⚠ cas où on cherche à <u>maximiser</u> la longueur.

L'idée est de se demander si $\pi_i + l_{ij} > \pi_j$
si la condition est vérifiée alors $\pi_j \leftarrow \pi_i + l_{ij}$
et $pred_j \leftarrow i$.

On répète le processus $n-1$ fois ou bien on s'arrête avant cad si $\forall ij \in U \;\; \pi_i + l_{ij} \leq \pi_j$

⚠ ce n'est pas la définition de π est un potentiel.

<u>Algorithme</u>

initialisation $\pi_i = \begin{cases} 0 & \text{si } i = s \\ l_{si} & \text{si } i \text{ successeur de } s \\ -\infty & \text{sinon} \end{cases}$

$pred_i = \begin{cases} s & \text{si } i = s \text{ ou } i \text{ successeur de } s \\ \phi & \text{sinon} \end{cases}$

$k \leftarrow 1$

Tant que $(k < n)$ et $(\forall ij \in U \;\; \pi_j \geq \pi_i + l_{ij})$

Pour tout $ij \in U$

Si $\pi_j < \pi_i + l_{ij}$

$\pi_j \leftarrow \pi_i + l_{ij}$

$pred_j \leftarrow i$

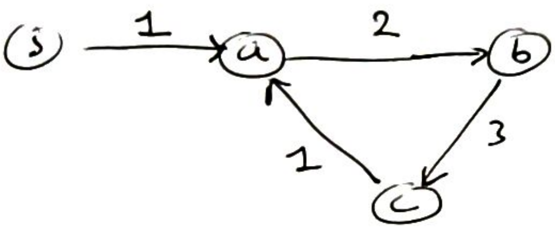
Fin Si

Fin Pour

Fin Tant que.

Rq si à la fin de l'algorithme $k = n-1$ et $\exists ij \in U$ $\pi_j > \pi_i + \ell_{ij}$ alors il y avait un circuit "grossissant" dans le graphe

Def circuit grossissant : $\sum_{u \in U} \ell(u) > 0$



plus long chemin de s à b :

sab : 3

sabcab = 10

$\vdots$

sabcabcab...cab = $+\infty$

Algorithme de Dijkstra

⚠ cas de la minimisation

On souhaite avoir le plus cours chemin vers les autres sommet à partir d'un sommet s.
L'idée est d'initialisée le potentiel du sommet s à 0 et les autres sommet à $+\infty$

Comme pour l'algorithme de Ford, on cherche à avoir la distance minimal à chaque passage d'un sommet à l'autre ie $\ell_{ij} + \pi_i < \pi_j$

dans ce cas $\pi_j \leftarrow \ell_{ij} + \pi_i$

$pred_j \leftarrow i$

Cependant, on n'applique pas ce principe sur tout les sommets

En partant du sommet s, on choisit le successeur qui possède le plus petit potentiel.
On vient ensuite placer ce successeur dans un ensemble $S = \{s\}$ et on l'enlève de $\overline{S} = \{a, b, c, d \dots\}$

À partir de ce nouveau sommet i, on met à jour les potentiels des successeurs de i qui ne sont pas dans S.
On répète le principe jusqu'à ce que $\overline{S} = \emptyset$

Algorithme

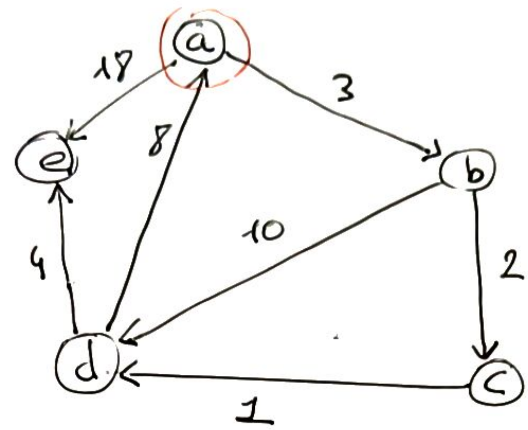
initialisation $\quad \pi_i = \begin{cases} 0 & \text{si } i = s \\ \ell_{si} & \text{si} \\ \infty & \text{sinon} \end{cases}$

$\text{pred}_i = \begin{cases} s & \text{si } i = s \text{ ou } i \text{ successeur de } s \\ \emptyset & \text{sinon} \end{cases}$

$S = \{s\} \qquad \overline{S} = X - \{s\}$

Tant que $\overline{S} \neq \emptyset$
$\quad j \leftarrow \underset{i \in \overline{S}}{\text{argmin}}\ \pi_i$
$\quad \overline{S} \leftarrow \overline{S} - \{j\}$
$\quad S \leftarrow S \cup \{j\}$
$\quad$ Si $\overline{S} \neq 0$ alors
$\quad\quad$ Pour tout $i \in R^+(j) \cap \overline{S}$
$\quad\quad\quad$ Si $\pi_i + \ell_{ij} < \pi_j$ alors
$\quad\quad\quad\quad \pi_j \leftarrow \pi_i + \ell_{ij}$
$\quad\quad\quad\quad \text{pred}_j \leftarrow i$
$\quad\quad\quad$ Fin Si
$\quad\quad$ Fin Pour
$\quad$ Fin Si
Fin Tant que

Exemple

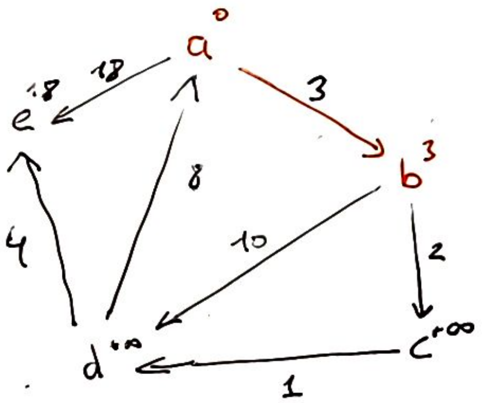


- On commence de a
  - ↳ mise à jour des potentiels des successeurs
  - ↳ choisit le sommet qui a le plus petit potentiel
  - ↳ mise à jour de S

initialisation

	a	b	c	d	e
Π	0	3	$+\infty$	$+\infty$	18
pred	a	a	$\emptyset$	$\emptyset$	a

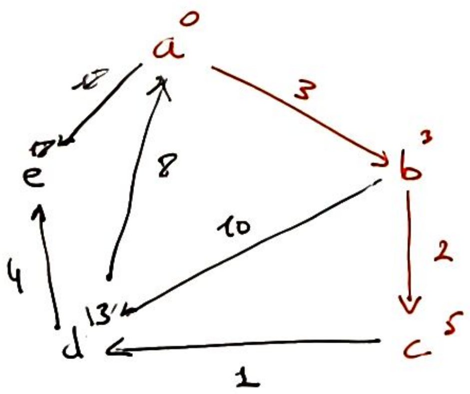
$\Rightarrow$ argmin = b



1ère boucle

	a	b	c	d	e
Π	0	3	5	13	18
pred	a	a	b	b	a

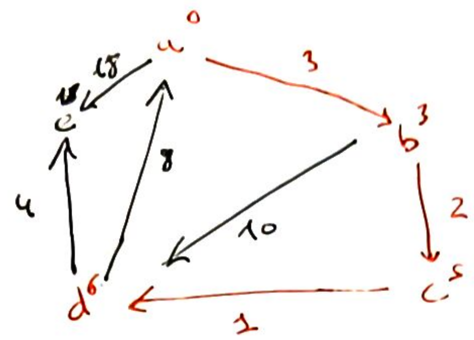
$\Rightarrow$ argmin = c



2ème boucle

	a	b	c	d	e
Π	0	3	5	6	18
pred	a	a	b	c	a

$\Rightarrow$ argmin = d



3ème boucle

	a	b	c	d	e
π	0	3	5	6	10
pred	a	a	b	c	d

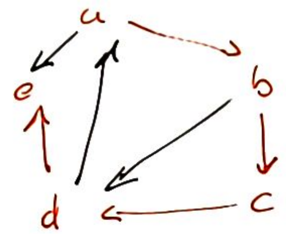
$\Rightarrow$ argmin $= e$

on place e dans S

donc $\bar{S} = \emptyset$

$\Rightarrow$ l'algo est fini.

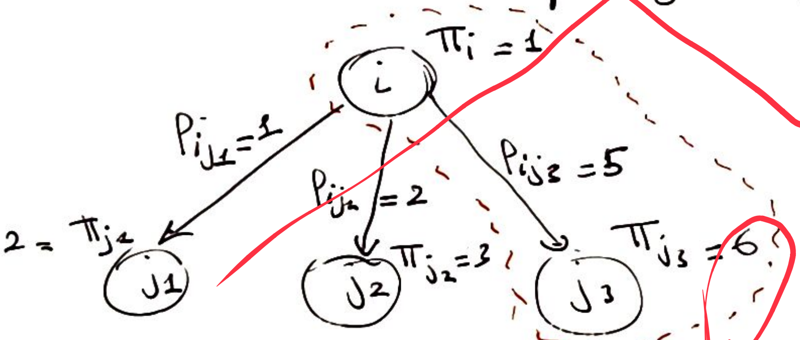
chemin les plus courts en partant de a :



Marche pas

⚠ cas de la maximisation de la longueur des chemins.

Comme pour Ford mais en appliquant le principe de Dikjstra, on se demande à chaque étape quel successeur donne le plus grand potentiel.

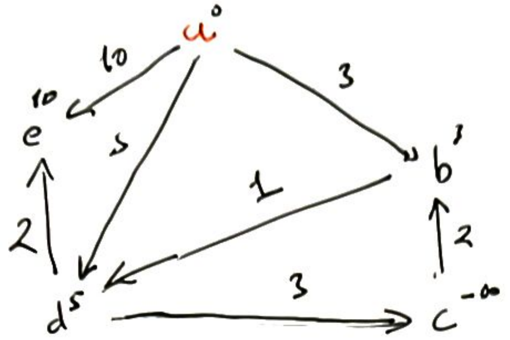


si $\pi_i + l_{ij} > \pi_j$ alors

$\pi_j \leftarrow \pi_i + l_{ij}$

$pred_j \leftarrow i$

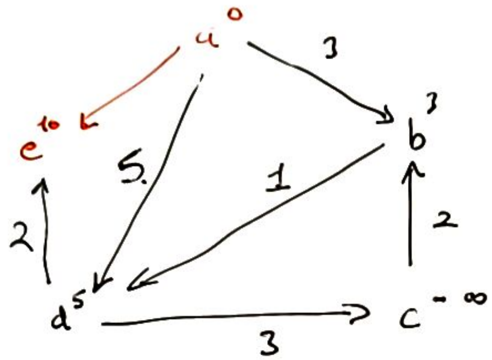
et à chaque étape on prend argmax des sommets non visités comme sommet successeur.



	a	b	c	d	e
π	0	3	$-\infty$	5	10
pred	a	a	$\emptyset$	a	a

$\Rightarrow$ argmax = e

$S = \{a, e\}$ $\overline{S} = \{b, c, d\}$



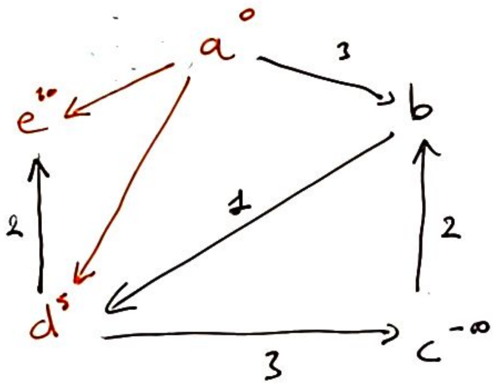
⚠ e n'a pas de successeur

Donc on ne rentre pas dans la boucle qui met les potentiels à jours.

Mais il y a toujours des sommets dans $\overline{S}$.

$\Rightarrow$ argmax = d

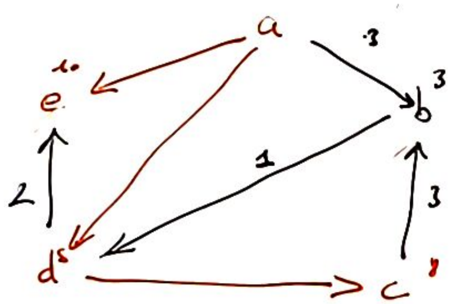
$S = \{a, e, d\}$ $\overline{S} = \{b, c\}$



	a	b	c	d	e
π	0	3	8	5	10
pred	a	a	d	a	a

$\Rightarrow$ argmax = c

$S = \{a, e, d, c\}$ $\overline{S} = \{b\}$



	a	b	c	d	e
π	0	6	8	5	10
pred	a	c	d	a	a

$\Rightarrow$ argmax = b

$S = \{a, e, d, c, b\}$ $\overline{S} = \emptyset$

Algo fini

chemin les plus long à partir de a :