

Chapitre 4

RECHERCHE DU PLUS COURT CHEMIN

DEFINITIONS

Réseau

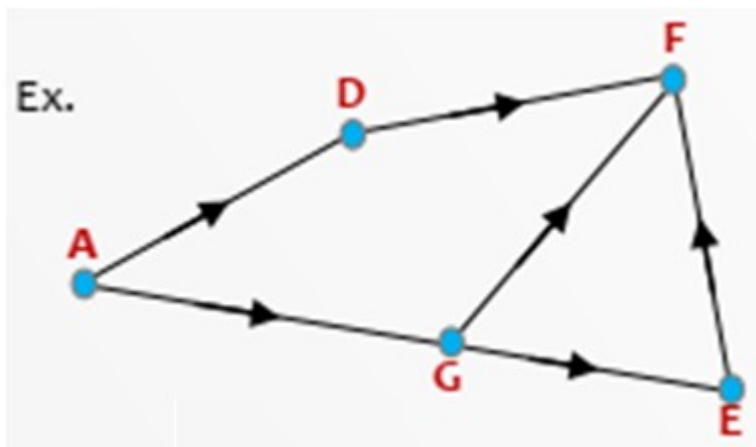
Un réseau est un graphe $G = (X; U)$ auquel on associe une fonction $d: U \rightarrow \mathbb{R}$ qui à chaque arc fait correspondre sa "longueur". On note $R = (X; U; d)$ un tel réseau.

Longueur

La longueur d'un chemin (d'une chaîne, d'un circuit ou d'un cycle) est la somme des longueurs de chaque arc qui le compose. Par convention, un chemin (une chaîne, un circuit ou un cycle) qui ne contient pas d'arc est de longueur nulle.

Ordre topologique (Organisation du graphe en niveaux):

Soit le graphe suivant:



- Le niveau (ou rang) d'un sommet Z est la longueur du plus long chemin d'extrémité terminale Z .

En considérant le dictionnaire des précédents $P(X)$, les sommets de niveau 0 sont ceux qui sont sans précédents. Le niveau 1 sera l'ensemble des sommets dont tous les précédents appartiennent au niveau 0, le niveau 2 l'ensemble des sommets dont tous les précédents appartiennent aux niveaux 0 et 1, et ainsi de suite. Le niveau sera noté par la lettre C .

Reprenons le dictionnaire des précédents de la figure 1.5 ci-dessus, pour voir comment définir les niveaux des sommets d'un digraphe :

Exemple 1.3.

X	P(X)
A	-
D	A
E	G
F	D, E, G
G	A

$$C_0 = \{A\}$$



X	P(X)
A	-
D	A
E	G
F	D, E, G
G	A

$$C_1 = \{D, G\}$$



X	P(X)
A	-
D	A
E	G
F	D, E, G
G	A

$$C_2 = \{E\}$$



X	P(X)
A	-
D	A
E	G
F	D, E, G
G	A

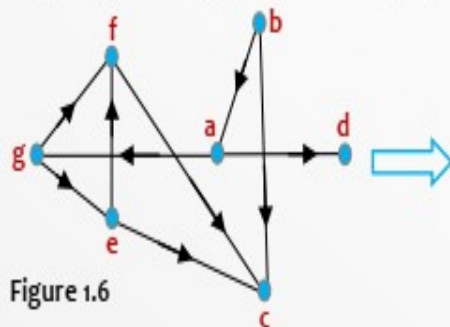
$$C_3 = \{F\}$$

Commentaire : A est sans précédents, il est donc de niveau 0 ; les sommets de niveau 1 sont trouvés en barrant d'un trait le sommet A dans toutes les cases où il apparaît, les sommets qui deviennent sans précédents après cette opération, constituent le niveau 1 [ici D et G] ; en suite en barrant D et G, partout où on les trouve, les sommets devenus sans précédents forment le niveau 2, ainsi de suite.

Note importante : La définition des niveaux des sommets n'est possible que pour un digraphe sans circuits. En effet, l'existence d'un circuit impliquerait une contradiction dans les précédences : une tâche devant en même temps précéder et succéder une autre !

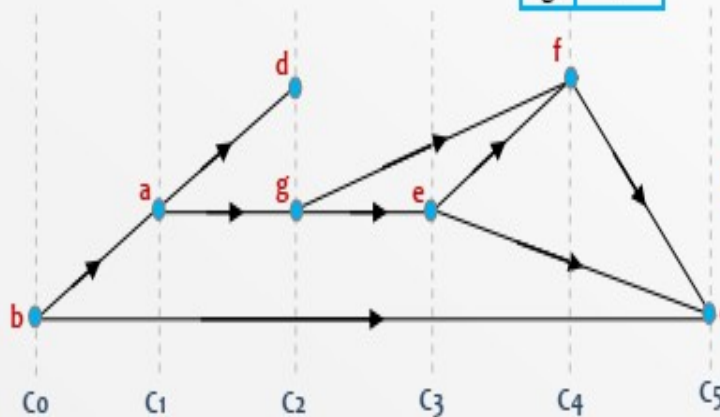
La notion des niveaux présente l'avantage de permettre l'ordonnancement du graphe à niveaux, c'est-à-dire de proposer une présentation plus simple du graphe, où les sommets sont placés de gauche à droite par ordre croissant des niveaux de sorte, si possible, qu'il n'y ait pas d'arcs qui se coupent.

Exemple 1.4 : Soit le graphe orienté ci-après. On demande de l'ordonner à niveaux.



X	P(X)
a	b
b	-
c	b, e, f
d	a
e	g
f	e, g
g	a

$C_0 = \{b\}$
 $C_1 = \{a\}$
 $C_2 = \{d, g\}$
 $C_3 = \{e\}$
 $C_4 = \{f\}$
 $C_5 = \{c\}$

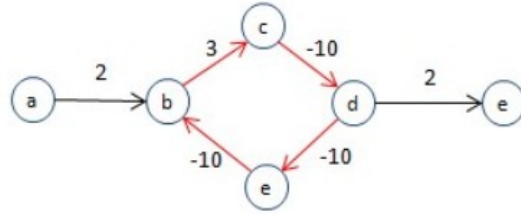


Est le graphe ordonné à niveaux.

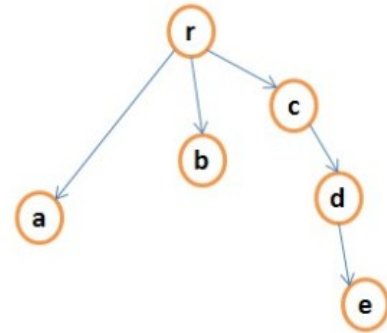
Note : Un graphe dont les sommets sont classés par niveau, est appelé **graphe ordonné à niveaux**. On parle également du **partage du graphe en niveaux**.

Circuit absorbant

- **Définition** : on appelle **circuit absorbant** un circuit de longueur négative

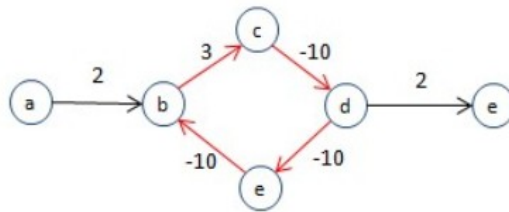


- **Définition** : on appelle **racine** du graphe orienté $G = (X, U)$ tout sommet de X vérifiant la propriété suivante : pour tout sommet y de X ($y \neq x$), il existe dans G un chemin de x à y .



- **Définition** : un graphe orienté $G = (X, U)$, avec $r \in X$, est une **arborescence de racine r** , si G est un arbre et r une racine de G

ex: $x = a$



- **Théorème** : une CNS pour que le problème 2 ait une solution est que x soit racine du graphe et que le graphe ne contienne pas de circuit absorbant
- Cas (faciles à résoudre) où l'on est sûr de l'absence de circuit absorbant :
 - si toutes les valuations sont positives ou nulles
 - si le graphe est sans circuit

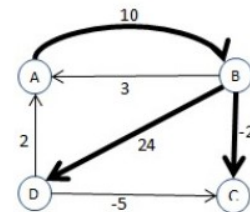
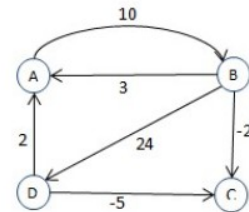
PROBLEMES DE PLUS (COURT/ LONG) CHEMIN

- Soit $G=(X,U)$ un graphe orienté valué
- $\ell : U \rightarrow \mathbb{R}$ fonction longueur, ou poids, ou coût, ou profit (Ex : $\ell(A,B)=10$)
- On définit la longueur d'un chemin de G comme la somme des longueurs de chacun des arcs qui le composent
- Ainsi étant donné un chemin μ de G :

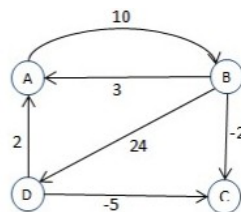
$$\mu = [u_1, u_2, \dots, u_p]$$

on définit **sa longueur** par :

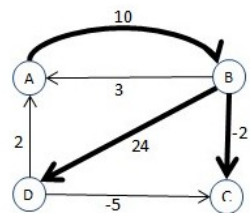
$$L(\mu) = \ell(u_1) + \ell(u_2) + \dots + \ell(u_p)$$



- **Problème 1** : étant donnés deux sommets x et y , trouver un chemin de longueur minimum de x à y



- **Problème 2** : étant donné un sommet x , trouver le plus court chemin de x à tous les sommets du graphe



- **Problème 3** : trouver, pour toute paire de sommets x et y , un plus court chemin de x à y

Donc, on peut s'intéresser à la recherche d'un plus court chemin dans un graphe :

- 1 - entre deux sommets donnés
- 2 - d'un sommet à tous les autres
- 3 - entre tous les couples de sommets

Rappelons que: Si un graphe possède un circuit absorbant, alors il n'existe pas de plus court chemin entre certains de ses sommets.

Théorème : Soit G un graphe orienté pondéré n'ayant pas de circuits absorbants, et x et y deux sommets de G . Si 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 . A

Propriété 1 : Tout sous-chemin d'un plus court chemin est un plus court chemin. **Propriété 2** : Si il existe un plus court chemin entre deux sommets x et y , alors il existe un plus court chemin élémentaire (sans cycle) entre x et y .

Propriété 2 : Si il existe un plus court chemin entre deux sommets x et y , alors il existe un plus court chemin élémentaire (sans cycle) entre x et y .

1°- Chemin optimal entre deux sommets donnés

Lester Randolph Ford, Jr (1927 -), mathématicien américain, a proposé un algorithme qui permet de calculer le chemin de valeur optimale [maximale ou minimale] allant d'un sommet choisi comme sommet initial à un sommet donné, dans un graphe orienté-valué et sans circuits.



Lester R. Ford (1927-)

L'algorithme peut s'énoncer ainsi : « un chemin de valeur optimale θ entre Z_1 et Z_n d'un graphe orienté est constitué des chemins de valeur optimale reliant deux sommets du chemin θ ».

L'algorithme est connu sous le nom de **l'Algorithme de Ford**.

L'on dira donc qu'un chemin θ allant d'un sommet Z_1 à un autre Z_n du digraphe est de valeur optimale [maximale ou minimale] si la somme des valuations $l(x_i, x_j)$ de chaque arc $x_i x_j = \theta_h$ est optimale [maximale ou minimale]. La longueur du chemin θ de valeur optimale sera $l(\theta) = \sum_{\theta_h \in \theta} l(\theta_h)$

Qu'il s'agisse de la recherche du plus long chemin [chemin de valeur maximale] ou du plus court [chemin de valeur minimale], la résolution par l'algorithme de Ford s'effectue principalement en 4 étapes :

Chemin de valeur maximale

1. Affecter un indice t_i **majorant** à chaque sommet du digraphe valué.

Poser $t_o = 0$, pour le sommet initial.

Poser $t_i = 0$, pour le reste des sommets [$\forall i \neq o$].

2. Les indices t_i sont successivement modifiés en choisissant un arc (X_i, X_j) s'il existe pour lequel la différence $e_{ji} = t_j - t_i < l(X_i, X_j)$.

Si $e_{ji} < l(X_i, X_j)$, t_j devient $t_j' = t_i + l(X_i, X_j)$. Sinon maintenir t_j , puis continuer à sonder d'autres arcs.

3. Quand il n'est plus possible, après un certain nombre d'ajustements, de modifier les t_i , les dernières valeurs portées sur chaque sommet donnent les longueurs des **chemins maximums** entre le sommet initial et le sommet terminal du **plus long chemin** recherché.

4. A la fin, on évalue tous les arcs aboutissant au sommet terminal du chemin recherché pour se rassurer qu'il n'y a plus de modification à faire.

Chemin de valeur minimale

1. Affecter un indice t_i **minorant** à chaque sommet du digraphe valué.

Poser $t_o = 0$, pour le sommet initial.

Poser $t_i = \infty$, pour le reste des sommets [$\forall i \neq o$].

2. L'indice t_i est successivement modifié en choisissant un arc (X_i, X_j) s'il existe pour lequel la différence $e_{ji} = t_j - t_i > l(X_i, X_j)$.

Si $e_{ji} > l(X_i, X_j)$, t_j devient $t_j' = t_i + l(X_i, X_j)$. Sinon maintenir t_j , puis continuer à sonder d'autres arcs.

3. Quand il n'est plus possible, après un certain nombre d'ajustements, de modifier les t_i , les dernières valeurs portées sur chaque sommet donnent les longueurs des **chemins minimums** entre le sommet initial et le sommet terminal du **plus court chemin** recherché.

4. A la fin, on évalue tous les arcs aboutissant au sommet terminal du chemin recherché pour se rassurer qu'il n'y a plus de modification à faire.

Note : Lorsque le graphe est partagé en niveaux, la recherche du chemin optimal sur ce graphe revient à rechercher le chemin optimal partant du sommet [ou de l'un des sommets] de niveau 0 au sommet [ou de l'un des sommets] du dernier niveau.

Considérons l'exemple ci-dessous, pour voir comment la recherche des chemins extrémaux se fait concrètement :

Exemple 1.6. On donne le graphe suivant :

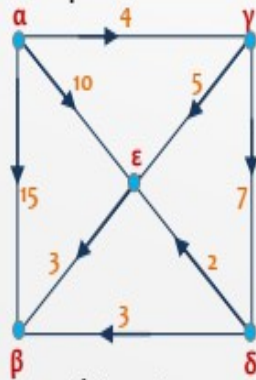


Figure 1.8

- On demande de rechercher sur le graphe ci-contre le plus long et le plus court chemin partant du sommet α au sommet β , en appliquant l'algorithme de Ford.

Recherche du plus long chemin

Posons $t_\alpha = 0, t_\beta = 0, t_\gamma = 0, t_\delta = 0, t_\epsilon = 0 \rightarrow$

❖ De α partent 3 arcs $(\alpha, \gamma), (\alpha, \beta)$ et (α, ϵ)

- (α, γ) $e_{\gamma\alpha} = t_\gamma - t_\alpha = 0 < 4 \rightarrow t_\gamma' = 0 + 4 = 4$
- (α, β) $e_{\beta\alpha} = t_\beta - t_\alpha = 0 < 15 \rightarrow t_\beta' = 0 + 15 = 15$
- (α, ϵ) $e_{\epsilon\alpha} = t_\epsilon - t_\alpha = 0 < 10 \rightarrow t_\epsilon' = 0 + 10 = 10$

❖ De γ partent 2 arcs (γ, ϵ) et (γ, δ)

- (γ, ϵ) $e_{\epsilon\gamma} = t_\epsilon' - t_\gamma' = 6 > 5 \rightarrow t_\epsilon' = 10$
- (γ, δ) $e_{\delta\gamma} = t_\delta - t_\gamma' = -4 < 7 \rightarrow t_\delta' = 4 + 7 = 11$

❖ De δ partent 2 arcs (δ, ϵ) et (δ, β)

- (δ, ϵ) $e_{\epsilon\delta} = t_\epsilon' - t_\delta' = -1 < 2 \rightarrow t_\epsilon' = 11 + 2 = 13$
- (δ, β) $e_{\beta\delta} = t_\beta' - t_\delta' = 4 > 3 \rightarrow t_\beta' = 15$

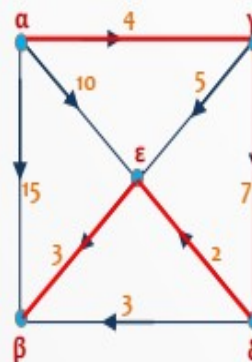
❖ De ϵ part 1 arc (ϵ, β)

- (ϵ, β) $e_{\beta\epsilon} = t_\beta' - t_\epsilon' = 2 < 3 \rightarrow t_\beta'' = 13 + 3 = 16$

❖ **Vérification** [on va sonder tous les arcs aboutissant directement à β]

- (α, β) $e_{\beta\alpha} = t_\beta'' - t_\alpha = 16 > 15 \rightarrow t_\beta'' = 16$
- (δ, β) $e_{\beta\delta} = t_\beta'' - t_\delta' = 5 > 3 \rightarrow t_\beta'' = 16$
- (ϵ, β) $e_{\beta\epsilon} = t_\beta'' - t_\epsilon' = 3 = 3 \rightarrow t_\beta'' = 16$

Le chemin le plus long recherché est $\theta = (\alpha\gamma\delta\epsilon\beta)$ et il mesure $l(\theta) = \sum_{\theta_h \in \theta} l(\theta_h) = 4 + 7 + 2 + 3 = 16$.



Recherche du plus court chemin

Posons $t_a = 0, t_b = \infty, t_\gamma = \infty, t_\delta = \infty, t_\epsilon = \infty$

❖ De a partent 3 arcs $(a, \gamma), (a, \beta)$ et (a, ϵ)

- (a, γ) $e_{\gamma a} = t_\gamma - t_a = \infty > 4 \rightarrow t_\gamma' = 0 + 4 = 4$
- (a, β) $e_{\beta a} = t_\beta - t_a = \infty > 15 \rightarrow t_\beta' = 0 + 15 = 15$
- (a, ϵ) $e_{\epsilon a} = t_\epsilon - t_a = \infty > 10 \rightarrow t_\epsilon' = 0 + 10 = 10$

❖ De γ partent 2 arcs (γ, ϵ) et (γ, δ)

- (γ, ϵ) $e_{\epsilon \gamma} = t_\epsilon' - t_\gamma' = 6 > 5 \rightarrow t_\epsilon'' = 4 + 5 = 9$
- (γ, δ) $e_{\delta \gamma} = t_\delta - t_\gamma' = \infty > 7 \rightarrow t_\delta' = 4 + 7 = 11$

❖ De δ partent 2 arcs (δ, ϵ) et (δ, β)

- (δ, ϵ) $e_{\epsilon \delta} = t_\epsilon'' - t_\delta' = -2 < 2 \rightarrow t_\epsilon'' = 9$
- (δ, β) $e_{\beta \delta} = t_\beta' - t_\delta' = 4 > \rightarrow t_\beta'' = 11 + 3 = 14$

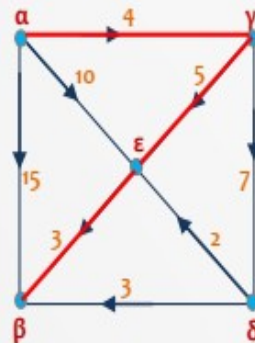
❖ De ϵ part 1 arc (ϵ, β)

- (ϵ, β) $e_{\beta \epsilon} = t_\beta'' - t_\epsilon'' = 5 > 3 \rightarrow t_\beta''' = 9 + 3 = 12$

❖ Vérification [on va sonder tous les arcs aboutissant directement à β]

- (a, β) $e_{\beta a} = t_\beta''' - t_a = 12 < 15 \rightarrow t_\beta''' = 12$
- (δ, β) $e_{\beta \delta} = t_\beta''' - t_\delta' = 1 < 3 \rightarrow t_\beta''' = 12$
- (ϵ, β) $e_{\beta \epsilon} = t_\beta''' - t_\epsilon'' = 3 = 3 \rightarrow t_\beta''' = 12$

Le chemin le plus court recherché est $\theta = (a\gamma\epsilon\beta)$ et il mesure $l(\theta) = \sum_{\theta_h \in \theta} l(\theta_h) = 4 + 5 + 3 = 12$.



Note : Le graphe devant être sans circuit pour appliquer l'algorithme, l'on devra veiller que le chemin optimal trouvé, maximal ou minimal, soit simple et élémentaire.

2°- D'un sommet à tous les autres

a)- Toutes les valuations sont positives ou nulles:

Algorithme de Moore-Dijkstra:

Principe:

- On étend une arborescence A , initialement réduite à la racine r
- À chaque itération, on étend l'arborescence par un nouvel arc et un nouveau sommet (son extrémité)
- À la fin de l'algorithme, l'arborescence construite donne pour chaque sommet x le plus court chemin de r à x

- L'algorithme fonctionne même en présence de circuit.

Algorithme:

(a) Initialisation

$$S = \{r = 1\}$$

$$S' = \{2, 3, \dots, n\}$$

$$\pi(1) = 0;$$

$$\pi(i) = +\infty \quad i = 2, 3, \dots, n$$

(b) **POUR** chaque sommet $j \in S'$ **FAIRE**

Chercher $J = \Gamma(j) \cap S$ et

$$\text{Calculer } \pi(j) = \min \left\{ \pi(j), \min_{k \in J} \{ \pi(k) + \omega_{kj} \} \right\}$$

(c) Sélectionner $j \in S'$ tel que $\pi(j) = \min_{i \in S'} \pi(i)$

- **Faire** $S \leftarrow S + \{j\}$
 $S' \leftarrow S' - \{j\}$

- **Si** $S' = \emptyset$ **Alors**

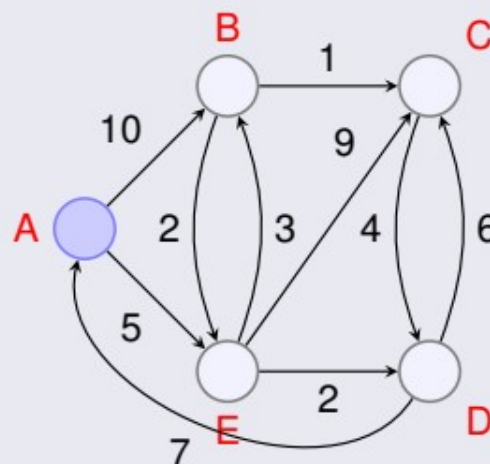
FIN

Sinon Aller à (b)

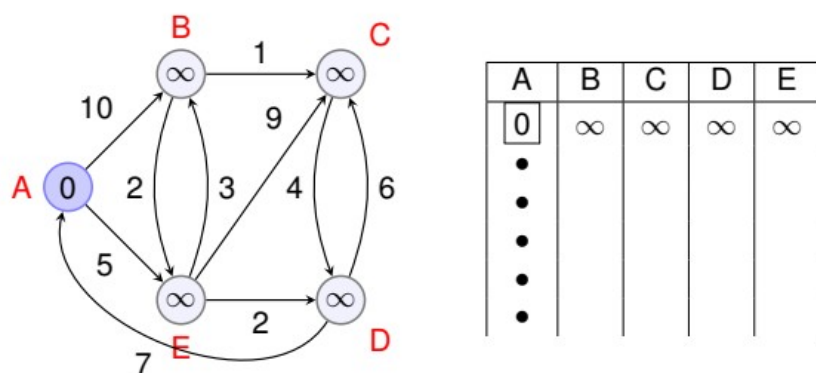
Fsi

Exemple 1

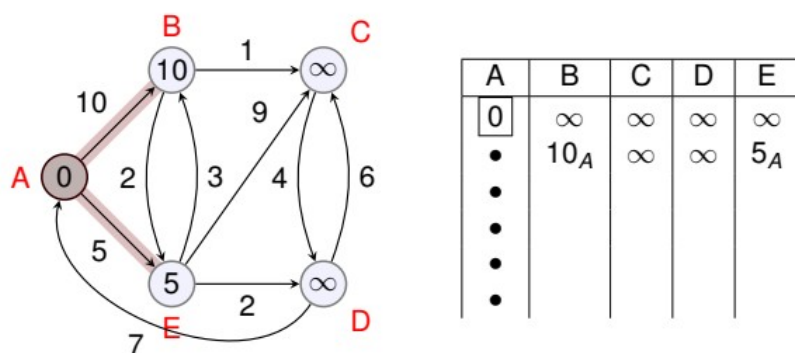
Cherchons les plus courts chemins d'origine A dans ce graphe:



On se place au sommet de plus petit poids, ici le sommet A.

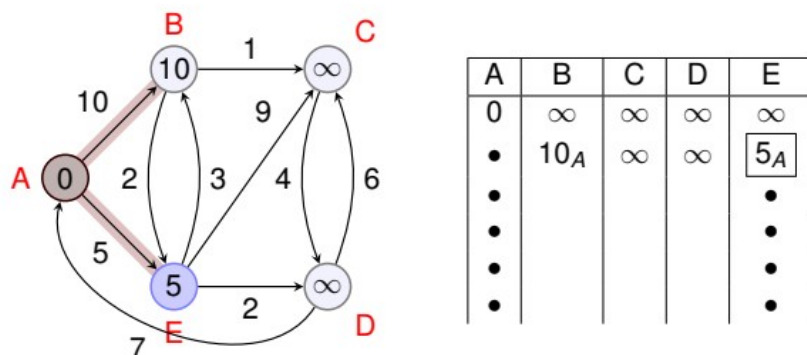


On étudie chacune des arêtes partant du sommet choisi.

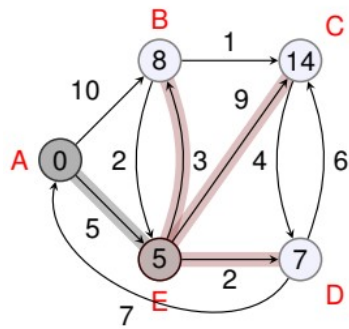


Dans les colonnes, on mets la distance à A, et le sommet d'où l'on vient.

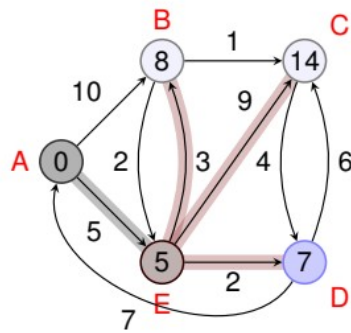
On se place de nouveau au sommet de plus petit poids, ici E.



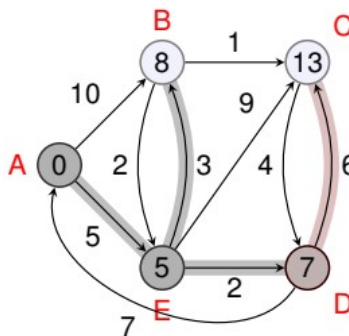
Et ainsi de suite.



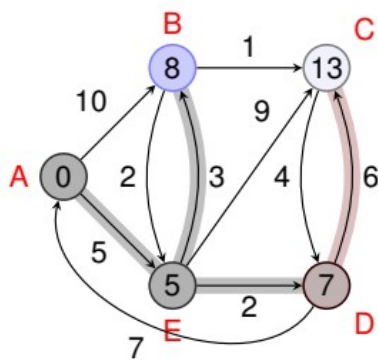
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•				•
•				•



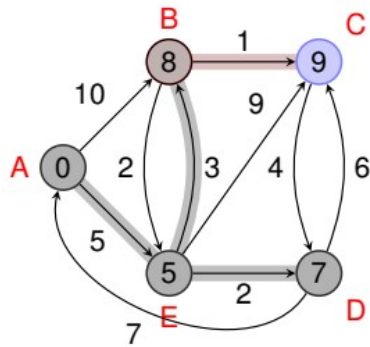
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•			•	•
•			•	•



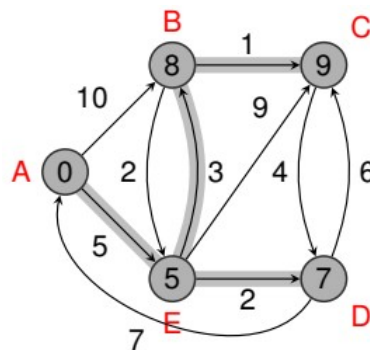
A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•	8_E	13_D	•	•
•			•	•
•			•	•



A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•	8_E	13_D	•	•
•	•		•	•
•	•		•	•



A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•	8_E	13_D	•	•
•	•	9_B	•	•
•	•	•	•	•



A	B	C	D	E
0	∞	∞	∞	∞
•	10_A	∞	∞	5_A
•	8_E	14_E	7_E	•
•	8_E	13_D	•	•
•	•	9_B	•	•
•	•	•	•	•

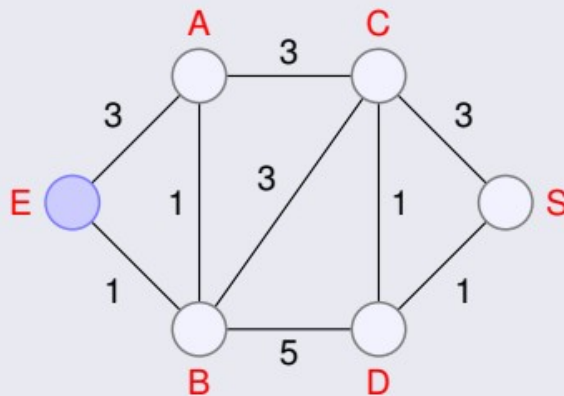
Si l'on ne considère que les flèches soulignées, on obtient un *arbre*, un graphe sans cycle.

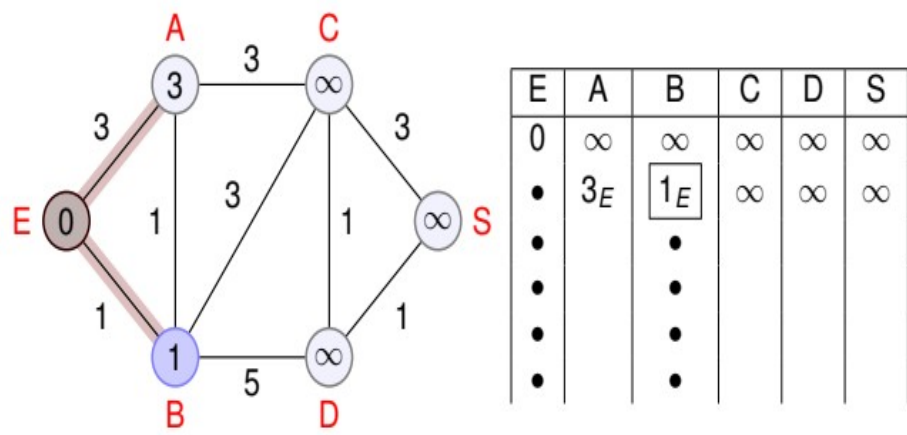
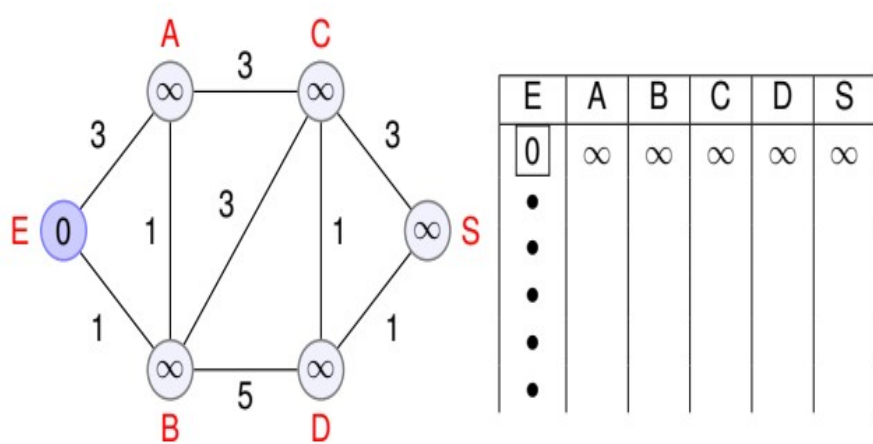
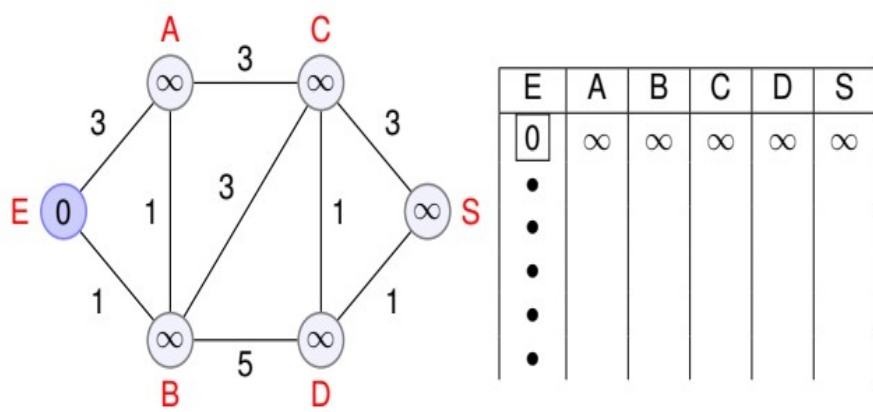
Arborescence des plus courts chemins:

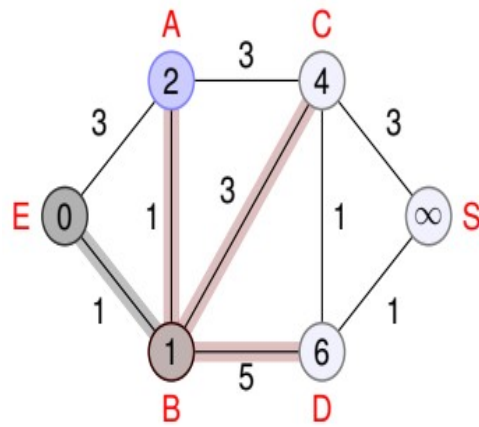
(Sur le dernier graphe, les sommets et les arcs en gris)

Exemple 2

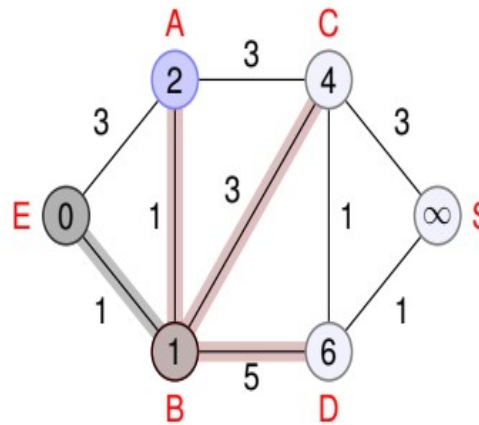
Cherchons les plus courts chemins d'origine *E* dans ce graphe:



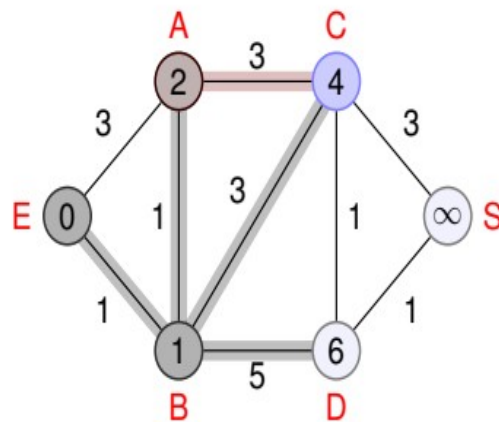




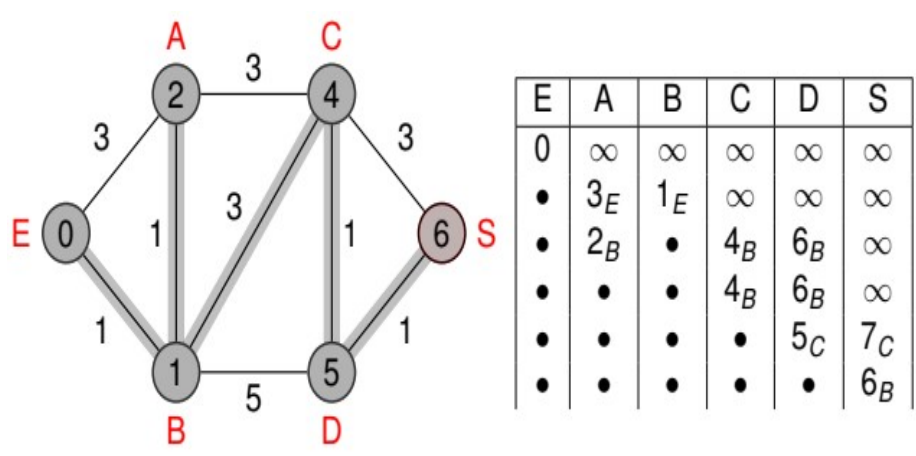
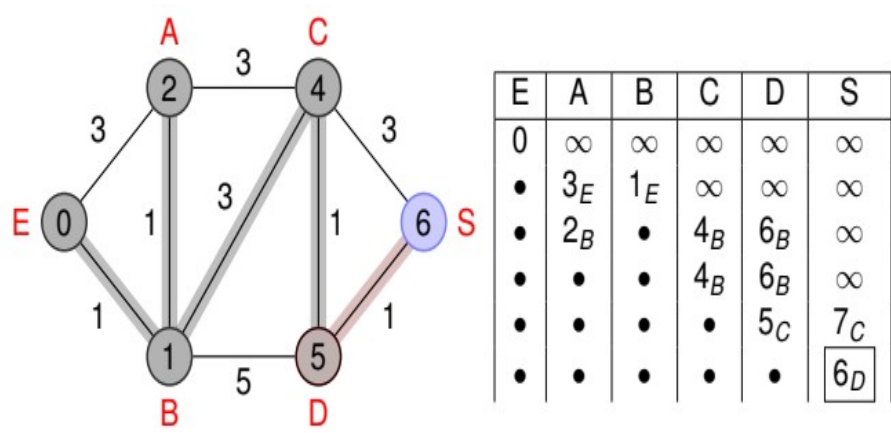
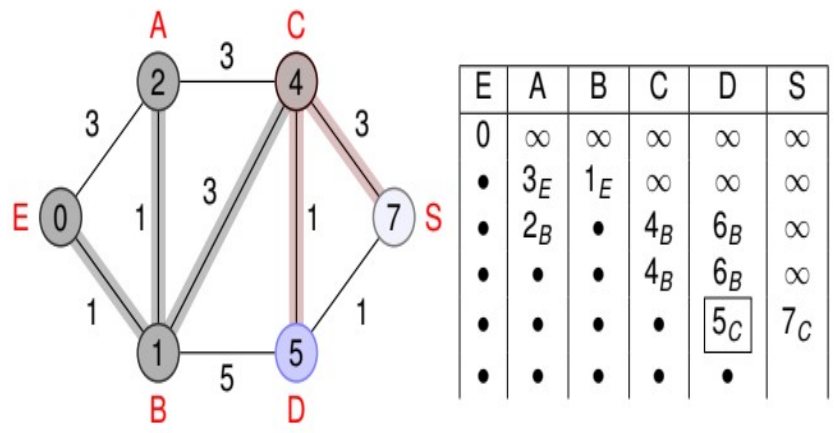
E	A	B	C	D	S
0	∞	∞	∞	∞	∞
•	3_E	1_E	∞	∞	∞
•	2_B	•	4_B	6_B	∞
•	•	•			
•	•	•			
•	•	•			



E	A	B	C	D	S
0	∞	∞	∞	∞	∞
•	3_E	1_E	∞	∞	∞
•	2_B	•	4_B	6_B	∞
•	•	•			
•	•	•			
•	•	•			



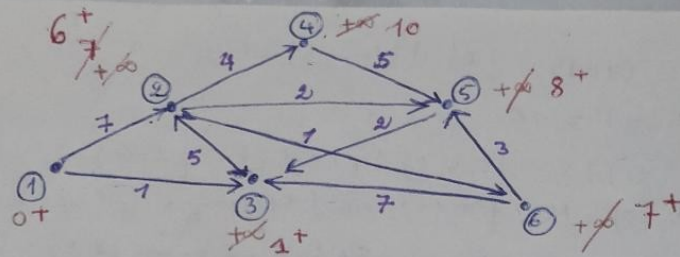
E	A	B	C	D	S
0	∞	∞	∞	∞	∞
•	3_E	1_E	∞	∞	∞
•	2_B	•	4_B	6_B	∞
•	•	•	4_B	6_B	∞
•	•	•	•		
•	•	•	•		



Arborescence des plus courts chemins:
 (Sur le dernier graphe, les sommets et les arcs en gris)

Exemple 3

Exp:
Dijkstra



$$S = \{1\} \quad \bar{S} = \{2, 3, 4, 5, 6\}$$

$$j = 2$$

$$J = \Pi^{-}(2) \cap S = \{1, 3\} \cap \{1\} = \{1\}$$

$$\begin{aligned} \pi(2) &= \min \{ \pi(2), \min \{ \pi(1) + w_{1,2} \} \} \\ &= \min \{ +\infty, \min \{ 0 + 7 \} \} = 7 \end{aligned}$$

$$j = 3$$

$$J = \Pi^{-}(3) \cap S = \{1, 5, 6\} \cap \{1\} = \{1\}$$

$$\pi(3) = \min \{ \pi(3), \min \{ \pi(1) + w_{1,3} \} \} = \min \{ +\infty, 1 \} = 1$$

$$j = 4$$

$$J = \Pi^{-}(4) \cap S = \{2\} \cap \{1\} = \emptyset$$

$$j = 5$$

$$J = \emptyset$$

$$j = 6$$

$$J = \emptyset$$

$$\text{selectionner } j \in \bar{S} \text{ tel } \pi(j) = \min_{i \in \bar{S}} \pi(i) \Rightarrow j = \{3\}$$

$$\text{donc } S = S + \{j\} = \{1\} + \{3\} \Rightarrow S = \{1, 3\}$$

$$\bar{S} = \{2, 4, 5, 6\}$$

$$j = 2$$

$$J = \Pi^{-}(2) \cap S = \{1, 3\} \cap \{1, 3\} = \{1, 3\}$$

$$\begin{aligned} \pi(2) &= \min \{ \pi(2), \min \{ \pi(1) + w_{1,2}, \pi(3) + w_{3,2} \} \} \\ &= \min \{ 7, \min \{ 0 + 7, 1 + 5 \} \} = \min \{ 7, \min \{ 7, 6 \} \} = 6 \end{aligned}$$

$$j = 4$$

$$\Pi^{-}(4) \cap S = \{2\} \cap \{1, 3\} = \emptyset$$

$$j = 5$$

$$J = \emptyset$$

$$j = 6$$

$$J = \emptyset$$

$$\text{selectionner } j = 2 \Rightarrow \bar{S} = \{1, 2, 3\}$$

$$\bar{S} = \{4, 5, 6\}$$

$$j = 4$$

$$\Pi^{-}(4) \cap S = \{2\} \cap \{1, 2, 3\} = \{2\}$$

$$\begin{aligned} \pi(4) &= \min \{ \pi(4), \min \{ \pi(2) + w_{2,4} \} \} \\ &= \min \{ +\infty, \min \{ 6 + 4 \} \} = 10 \end{aligned}$$

$$j = 5$$

$$\Pi^{-}(5) \cap S = \{2, 4, 6\} \cap \{1, 2, 3\} = \{2\}$$

$$\begin{aligned} \pi(5) &= \min \{ \pi(5), \min \{ \pi(2) + w_{2,5} \} \} \\ &= \min \{ +\infty, \min \{ 6 + 2 \} \} = 8 \end{aligned}$$

$$j = 6$$

$$\Pi^{-}(6) \cap S = \{2\} \cap \{1, 2, 3\} = \{2\}$$

$$\pi(6) = \min \{ \pi(6), \min \{ \pi(2) + w_{2,6} \} \} = \min \{ +\infty, \min \{ 6 + 1 \} \} = 7$$

$$\text{selectionner } \{6\} \text{ donc } S = \{1, 2, 3, 6\}$$

$$\bar{S} = \{4, 5\}$$

$j=4 \quad \Gamma^-(4) \cap S = \{2\} \cap \{1, 2, 3, 6\} = \{2\}$
 $\pi(4) = 10$

$j=5 \quad \Gamma^-(5) \cap S = \{2, 4, 1\} \cap \{1, 2, 3, 6\} = \{2, 6\}$
 $\pi(5) = \min \{ \pi(5), \min \{ \pi(2) + w_{25}, \pi(6) + w_{65} \} \}$
 $= \min \{ 8, \min \{ 6 + 2, 7 + 3 \} \} = \min \{ 8, 8 \} = 8$

sélectionner $\{5\}$
 $S = \{1, 2, 3, 5, 6\}$
 $\bar{S} = \{4\}$

$j=4 \quad \Gamma^-(4) = \{2\} \cap \{1, 2, 3, 5, 6\} = \{2\}$
 $\pi(4) = \min \{ \pi(4), \min \{ \pi(2) + w_{24} \} \} = \min \{ 10, \min \{ 6 + 4 \} \} = 10$

sélectionner $\{4\}$
 $S = \{1, 2, 3, 4, 5, 6\}$
 $\bar{S} = \{\emptyset\}$

donc

Algorithme de Bellman

Graphe sans circuit, valuations quelconques. Il est souhaitable que le graphe soit organisé en niveaux (racine = α , puits = ω).

Algorithme

1- $\pi(\alpha) = 0$;

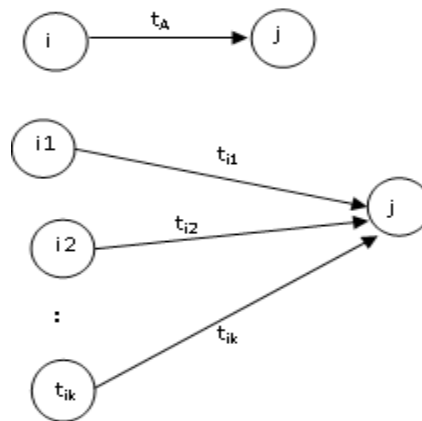
2- **POUR** chaque sommet i

S'il existe un seul arc (i, j) Alors

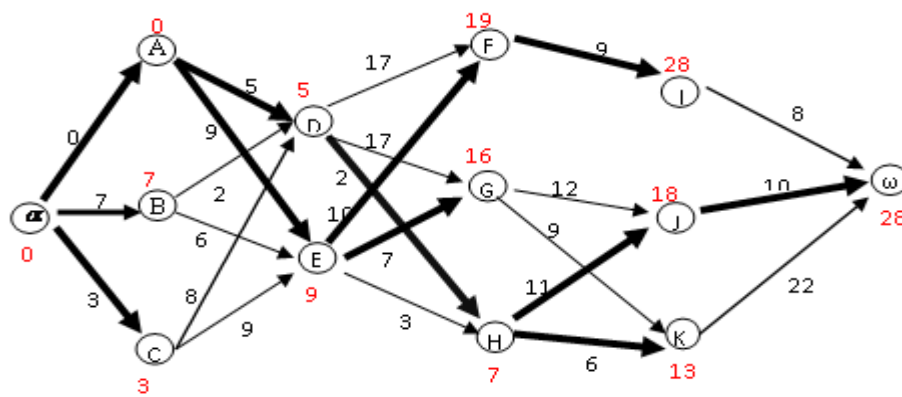
$\pi(j) = \pi(i) + t_{ij}$ (t_{ij} : valeur inscrite sur l'arc (i, j))

Sinon $\pi(j) = \min \{ \pi(i_1) + t_{i_1}, \pi(i_2) + t_{i_2}, \dots, \pi(i_k) + t_{i_k} \}$

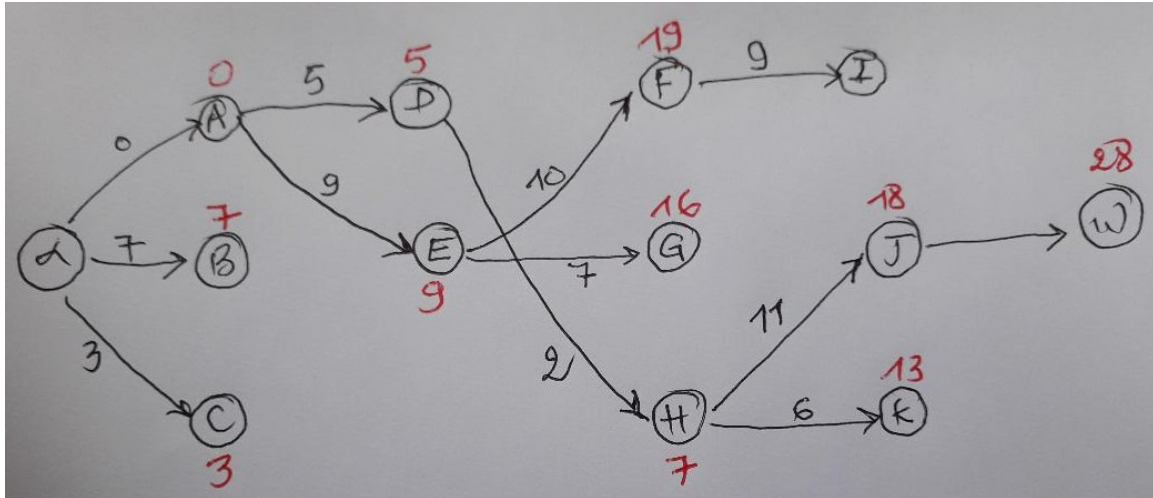
3- Les valeurs sur chaque sommet représente la valeur du plus court chemin du sommet racine à ce sommet.



Exemple



L'arborescence des plus courts chemins:



Remarque:

Dans le cas du plus long chemin, il suffit de changer **Min** par **Max** dans l'algorithme.