Introduction à la Théorie des Graphes

--

Partie 5 : Chemins Optimaux

Dr. Guelzim ibrahim
Email: ib.guelzim@gmail.com

Sommaire

Graphes valués : Introduction

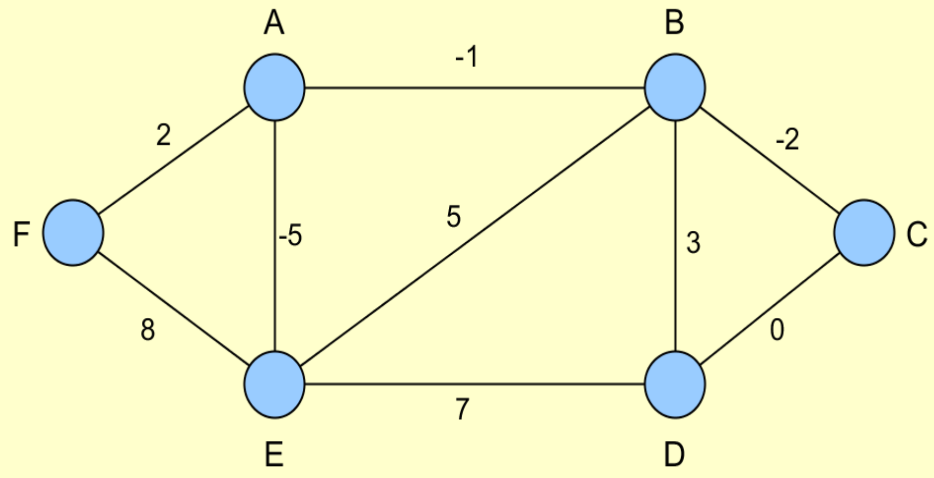
- Problème
  - Dans un graphe orienté G(V,E), trouver les plus courts chemins à partir d'un sommet de départ $s \in V$ vers tous les autres sommets.
- Pourquoi s'intéresse t-on à ces problèmes ?
  - Problèmes de tournées
  - Ordonnancement
  - Routage
  - ...
- Deux algorithmes
  - Algorithme de Dijkstra
  - Algorithme de Bellman-Ford

Graphes valués

- Définition
  - Soit $G = (V , E)$ un graphe orienté ou non.
  - On dit que G est valué si l'on attribue à chacun de ses arcs (ou arêtes) une valeur numérique.
  - Un graphe est dit à valuations positives si toutes ces valeurs sont positives ou nulles.
  - Dire que G est valué revient à dire qu'il existe une application v appelée valuation, définie sur E et à valeurs réelles : $$\begin{array}{cccc} V : & E & \rightarrow & R \\ & (x, y) & \rightarrow & v(x, y) \end{array}$$
- Si le graphe est non orienté, l'application valuation est symétrique

Graphes valués

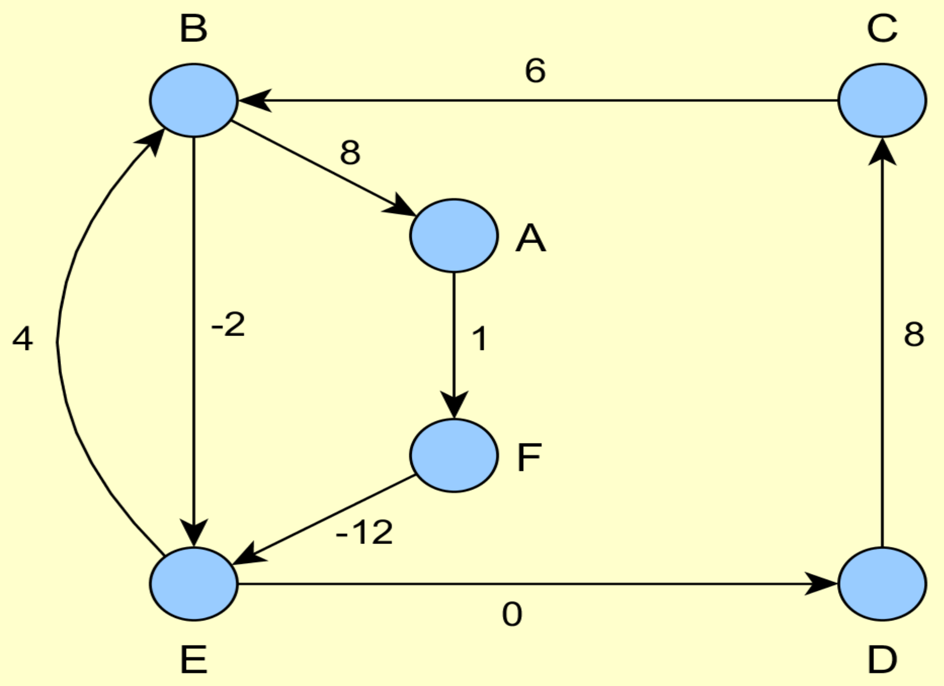
- Exemple : Considérons le graphe G non orienté suivant



- Ce graphe est valué, car chacune de ses arêtes possède une valeur réelle.
- En ce qui concerne l'application valuation, on a par exemple
  - v (A, B) = v (B, A) = −1.

Graphes valués

- Considérons le graphe G orienté suivant :



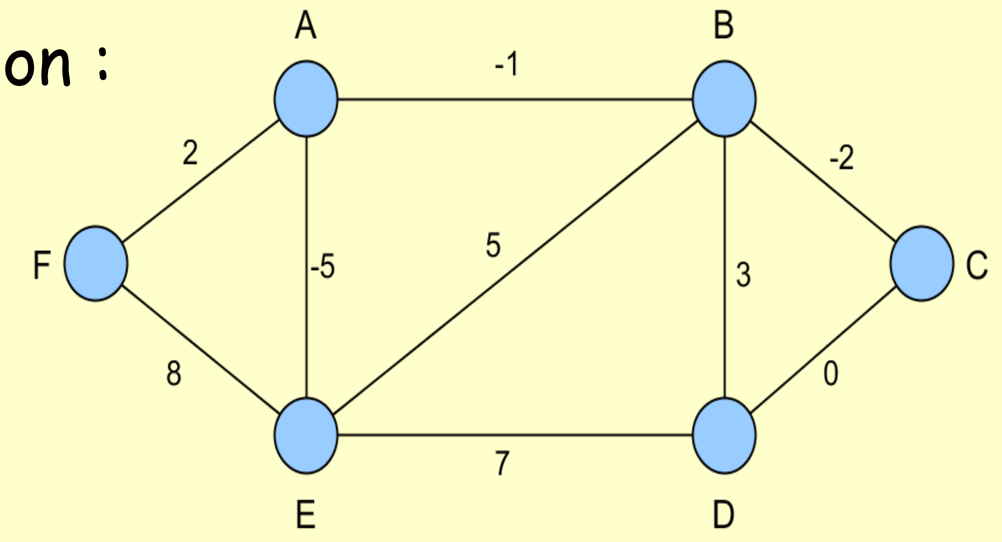
- En ce qui concerne l'application valuation, on a par exemple
    - v (B, E) = −2 et v (E, B) = 4

Graphes valués : Représentation matricielle

- Défintion:
  - Soit G = (V , E) un graphe valué orienté ou non d'ordre n dont on a numéroté les n sommets.
  - On peut le représenter par sa matrice de valuation, qui est une matrice carrée d'ordre n, comme suit :
    - $m_{ij} = v(i, j)$ si $(i, j) \in E$
    - $m_{ij} = +\infty$ sinon
- Contrairement à la matrice d'adjacence, on ne peut pas utiliser la valeur 0 pour indiquer qu'il n'y a pas d'arc (resp. d'arête) entre deux sommets.
- Cette valeur signifiera au contraire qu'il y a un arc (resp. une arête) et que la valuation de celui-ci (resp. celle-ci) vaut 0.
- Notre but étant de déterminer des plus courts chemin, nous utiliserons donc la valeur $+\infty$ dans ce cas, afin de ne pas interférer avec notre recherche.

Graphes valués : Représentation matricielle

- Ex Graphe Non orienté, sa matrice de valuation :

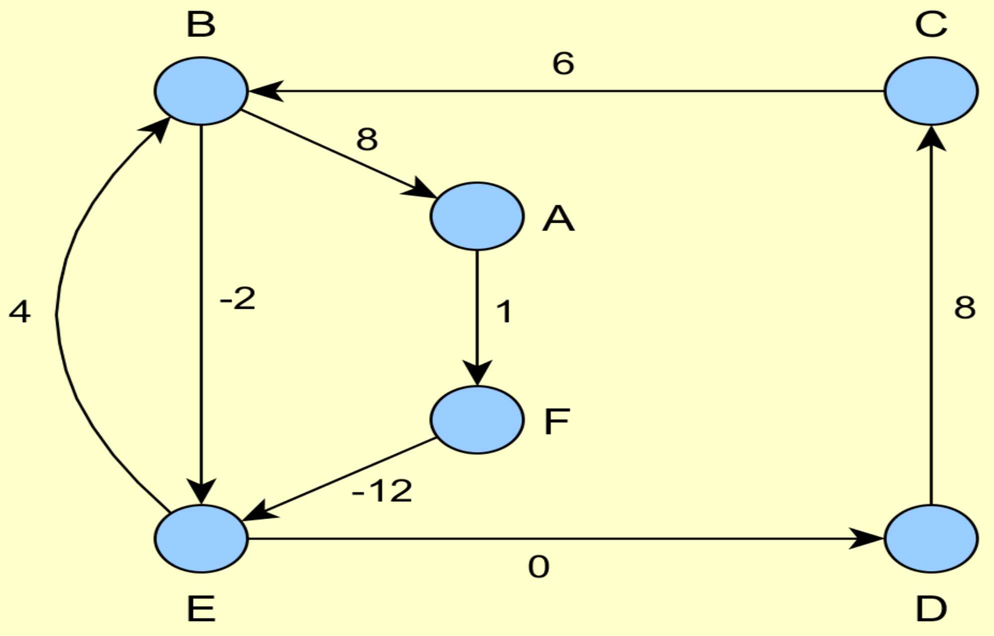


$$M_V = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} +\infty & -1 & +\infty & +\infty & -5 & 2 \\ -1 & +\infty & -2 & 3 & 5 & +\infty \\ +\infty & -2 & +\infty & 0 & -5 & +\infty \\ +\infty & 3 & 0 & +\infty & 7 & +\infty \\ -5 & 5 & -5 & 7 & +\infty & 8 \\ 2 & +\infty & +\infty & +\infty & 8 & +\infty \end{bmatrix} \end{array}$$

- Par exemple, la valeur 7 située à l'intersection de la 5ème ligne et de la 4ème colonne signifie que les sommets E et D sont adjacents et que la valuation de l'arête les reliant est égale à v(E , D) = v(D , E) = 7

Graphes valués : Représentation matricielle

- Ex Graphe orienté G :
- sa matrice de valuation :



$$M_V = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} +\infty & +\infty & +\infty & +\infty & +\infty & 1 \\ 8 & +\infty & +\infty & +\infty & -2 & +\infty \\ +\infty & 6 & +\infty & +\infty & +\infty & +\infty \\ +\infty & +\infty & 8 & +\infty & +\infty & +\infty \\ +\infty & 4 & +\infty & 0 & +\infty & +\infty \\ +\infty & +\infty & +\infty & +\infty & -12 & +\infty \end{bmatrix} \end{array}$$

- ○ $M_V(5,2) = M_V(E,B) = v(E,B) = 4$
- ○ $M_V(2,5) = M_V(B,E) = v(B,E) = -2$ (n'est pas symétrique)

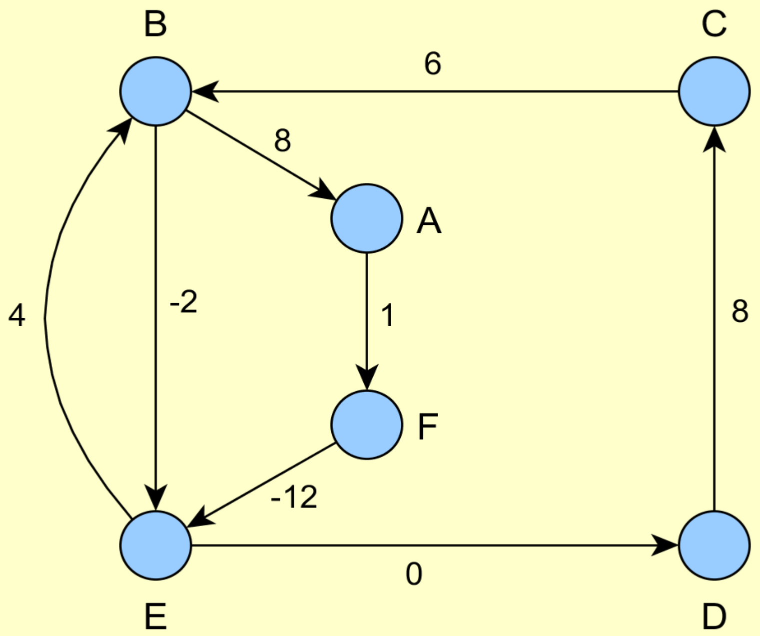
Graphes valués : Valuation d'un parcours

- Définition
  - Soit $G=(V,E)$ un graphe valué orienté (resp. non orienté).
  - La **valuation d'un chemin** (resp. d'une chaîne) est la **somme** des valuations des arcs (resp. arêtes) le constituant.
  - La valuation d'un chemin (resp. d'une chaîne) réduit à un sommet est **nulle**.

- Ces définitions à propos des chaînes et chemins s'appliquent aussi aux cas particuliers des cycles et circuits.

Graphes valués : Valuation d'un parcours

- Considérons le graphe valué orienté suivant :



- La valuation du chemin (B,E,B,A,F) est égale à : -2+4+8+1, c'est-à-dire 11
- La valuation du chemin (C) est égale à 0

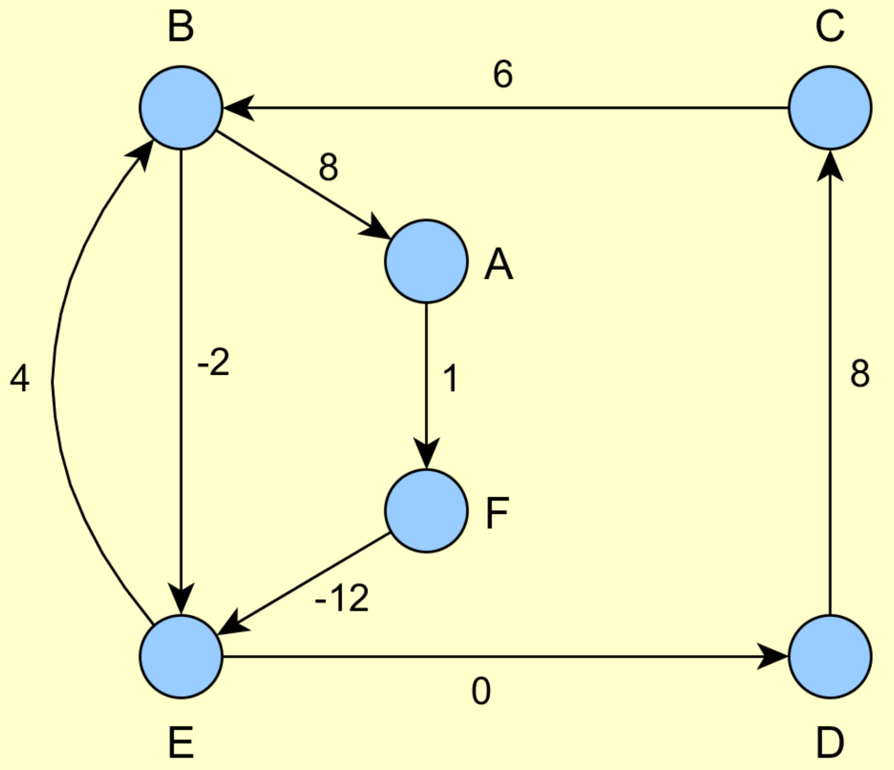
Graphes valués : Plus court chemin

- Définition
  - Soit $G = (V, E)$ un graphe valué orienté (resp. non orienté).
  - Soient x et y deux de ses sommets.
  - Si l'ensemble des valuations des chemins (resp. chaînes) d'origine x et d'extrémité y admet un **minimum**, celui-ci est appelé **distance** de x à y et est noté $d(x, y)$.
  - Dans ce cas, un **plus court chemin** (resp. **plus courte chaîne**) de x à y sera un chemin (resp. une chaîne) dont la valuation est égale à $d(x, y)$.

- Rq : Ce minimum n'existe pas forcément, comme nous le verrons par la suite.

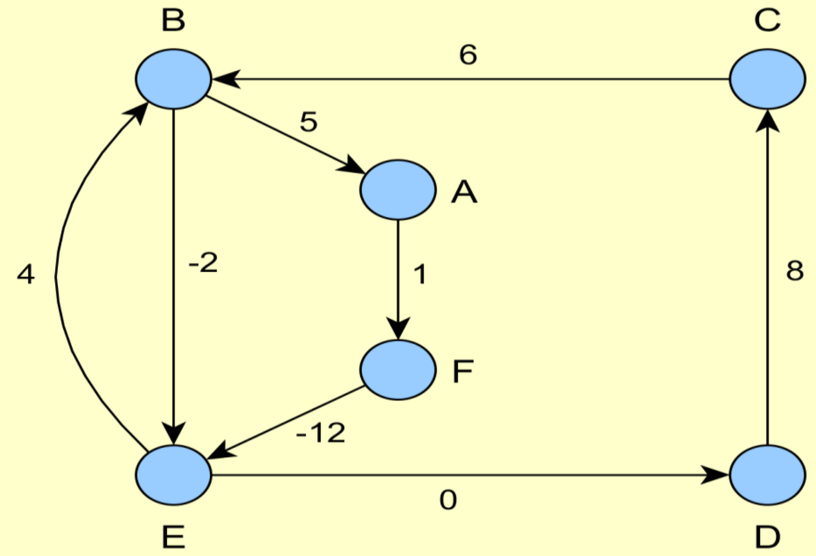
Graphes valués : Plus court chemin

- Exemple :
  - Considérons le graphe valué orienté suivant



- La distance de B à E vaut -3, et correspond au (plus court) chemin (B,A,F,E).

Graphes valués : Plus court chemin



- Considérons le graphe valué orienté ci-contre :
  - La valuation du chemin d'origine B et d'extrémité E défini par (B,A,F,E) est égale à -6
  - La valuation du circuit (E,B,A,F,E) est égale à -2
  - Si l'on parcourt ce circuit après le chemin on obtient un nouveau chemin d'origine B et d'extrémité E défini par (B,A,F,E,B,A,F,E) de valuation égale à -8
  - En parcourant ensuite de nouveau le circuit précédent, on obtient un chemin (B,A,F,E,B,A,F,E,B,A,F,E) de valuation égale à -10.
  - On peut répéter ce procédé autant de fois que l'on veut, dès que l'on effectuera un circuit supplémentaire, la valuation du chemin diminuera de 2.
  - L'ensemble des valuations des chemins d'origine B et d'extrémité E n'a donc pas de minimum
  - Il n'y a ainsi pas de plus court chemin allant de B à E.

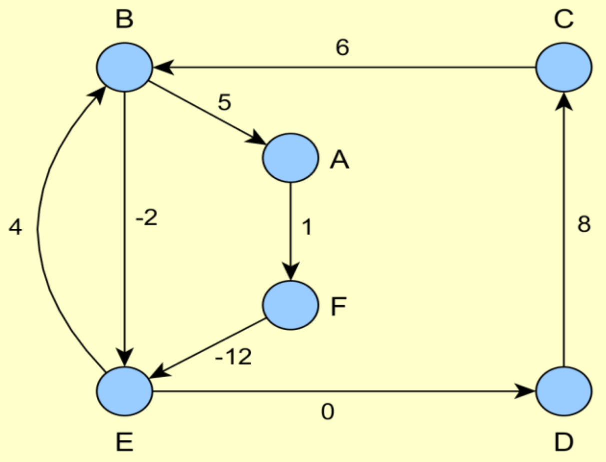
Graphes valués : Plus court chemin

- Proposition
  - Soit $G = (V; E)$ un graphe valué orienté (resp. non orienté).
  - Etant donné deux sommets x et y, il y a trois possibilités :
    1. Il existe un ou plusieurs plus courts chemins (resp. chaînes) de x à y.
    2. Il n'existe pas de chemins (resp. chaînes) de x à y.
    3. Il existe des chemins (resp. chaînes) de x à y mais pas de plus court.

- Rq :
  - Le 1er cas correspond à une distance finie entre x et y.
  - Le 2ème et 3ème cas, correspondent à une distance qu'on appelera par abus **distance infinie**.

Graphes valués : Plus court chemin

- Proposition
  - Soit $G = (V; E)$ un graphe valué orienté (resp. non orienté).
  - Un <u>**circuit**</u> (resp. cycle) <u>**absorbant**</u> de G est un circuit (resp. cycle) de valuation **strictement négative**.
- Exemple : Le circuit (E,B,A,F,E) dans le graphe de l'exemple ci-contre est un circuit absorbant.



- Théorème
  - Soit $G = (V, E)$ un graphe valué orienté (resp. non orienté).
  - On suppose que G ne possède pas de circuit (resp. cycle) absorbant.
  - Si entre deux sommets fixés du graphe il existe un chemin (resp. une chaîne), alors entre ces deux sommets il existe un plus court chemin (resp. plus courte chaîne).

Algorithme de Dijkstra : Principe

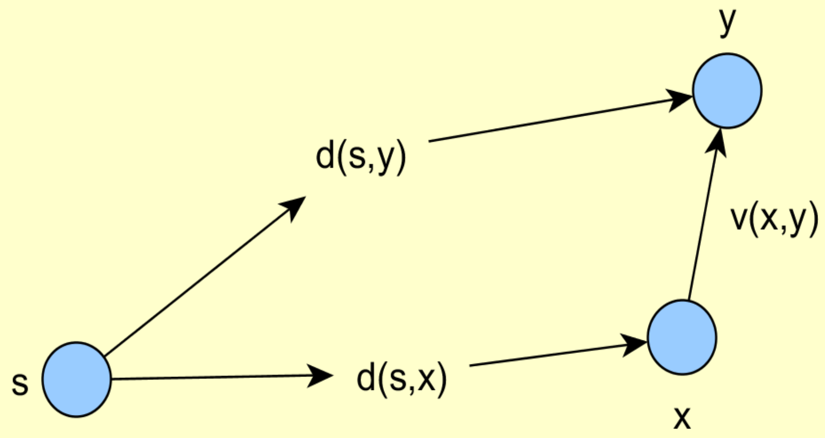
- L'algorithme de Dijkstra est un algorithme de recherche de plus court chemin entre un sommet fixé **s** et tous les autres sommets d'un graphe à valuations v **positives**.
- L'algorithme consiste en 2 phases,
  1. Initialisation
  2. Traitement (itérations)
- Si **n** est l'ordre du graphe, après une phase d'initialisation, cet algorithme procède en **n – 1** itérations (une itération / chaque sommet $x \neq s$).
- Initialisation :
  - Attribuer au sommet s une distance nulle, $d(s , s) = 0$
  - Attribuer à chaque sommet $x \neq s$ du graphe une distance provisoire = $v(s , x)$

Algorithme de Dijkstra : Principe

- L'initialisation citée ci-haut consiste à :
  1. Attribuer au sommet s une distance nulle, $d(s\ ,\ s) = 0$
  2. Attribuer à chaque sommet x successeur de s une distance provisoire $d(s\ ,\ x) = v(s\ ,\ x)$
  3. Les autres sommets n'étant pas reliés directement à s, on initialise leur distance avec la valeur $+\infty$

Algorithme de Dijkstra : Principe

- Traitement (itérations) :
  1) Choisir un sommet x parmi ceux qui <u>n'ont pas encore été traités</u>, dont la distance à s est **minimale**.
  2) Pour chacun des successeurs y de x, si $d(s, y) > d(s , x) + v(x, y)$ on met à jour la distance $d(s, y) \leftarrow d(s , x) + v(x, y)$
  3) Quand tous les successeurs du sommet x ont été examinés, on rajoute x à la liste des sommets traités,
  4) S'il existe un sommet non traité, retour à 1)



Algorithme de Dijkstra : Principe

- Algorithme de Dijkstra
  - Soit G = (V, E) un graphe orienté (ou non) à valuations <u>**positives**</u> d'ordre n.
  - Soit **s** le sommet à partir duquel on va rechercher les plus courts chemins.
  - Soit **L** un tableau de n cases destiné à contenir les distances de s aux autres sommets.
  - Soit **P** un tableau de n cases destiné à contenir le prédécesseur de chacun des sommets dans un plus court chemin d'origine s.
  - Soit **M** la liste de tous les sommets déjà traités (et donc V \ M sera la liste de tous les sommets à traiter).

Algorithme de Dijkstra : Principe

- Soit $G = (V, E)$ un graphe orienté (ou non) d'ordre n à valuations positives.

Initialisation:

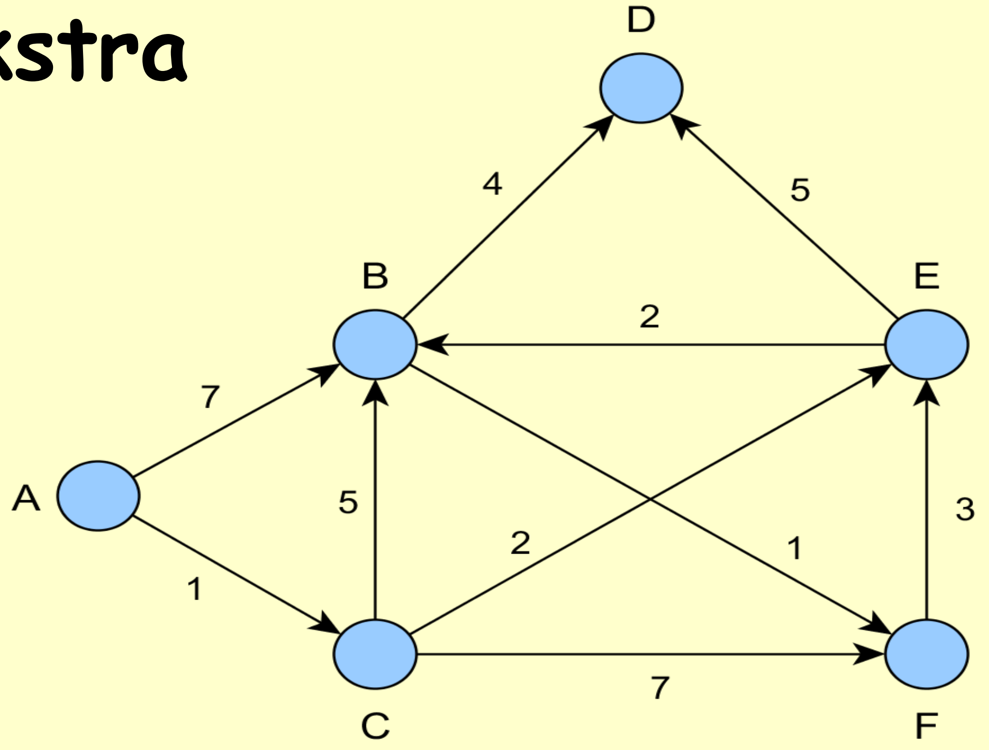
```
M ← {s}
L[s] ← 0
P[s] ← s
Pour tout x ≠ s
   L[x] ← v(s , x)
   Si x est un successeur de s
        P[x]  ← s
   Sinon
        P[x] ← Ø
   FinSi
FinPour
```

Traitement:

```
TantQue M ≠ V  Faire
choisir x ∈ V\M  tq ∀ y ∈ V\M, L[x] ≤ L[y]
        Pour Tout y ∈ V\M  tq y soit un succ de x
                Si L[y] > L[x] + v(x, y)
                      L[y] ← L[x] + v(x, y)
                      P[y] ← x
                Finsi
        Finpour
M ← M ∪ {x}
FinTantQue
```

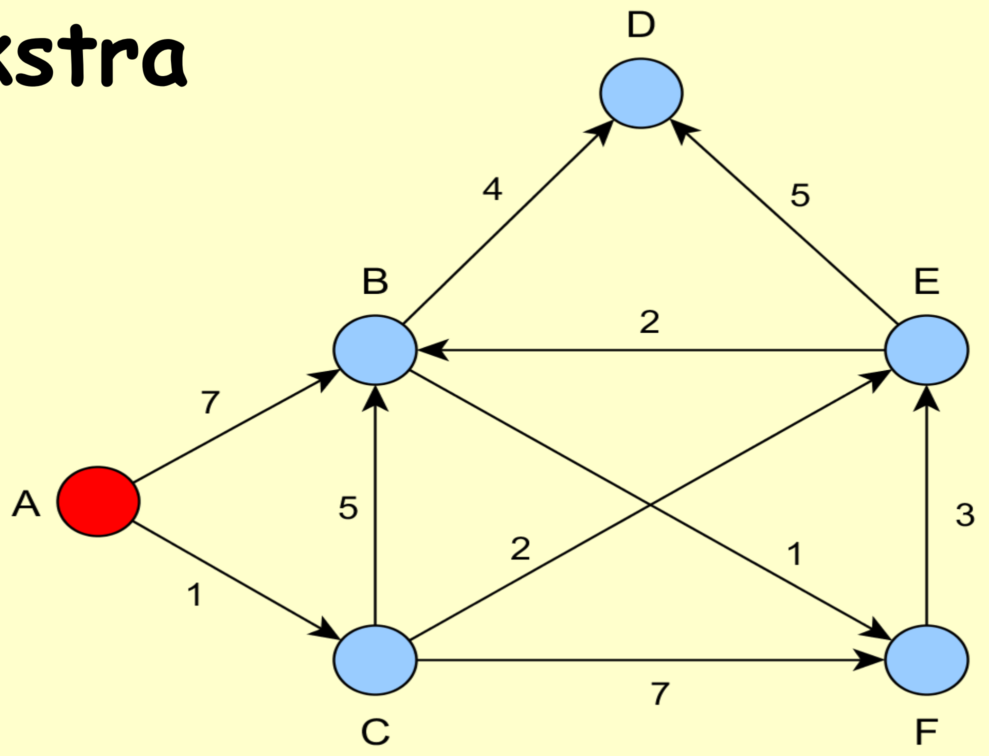
Algorithme de Dijkstra

- Exemple



Algorithme de Dijkstra

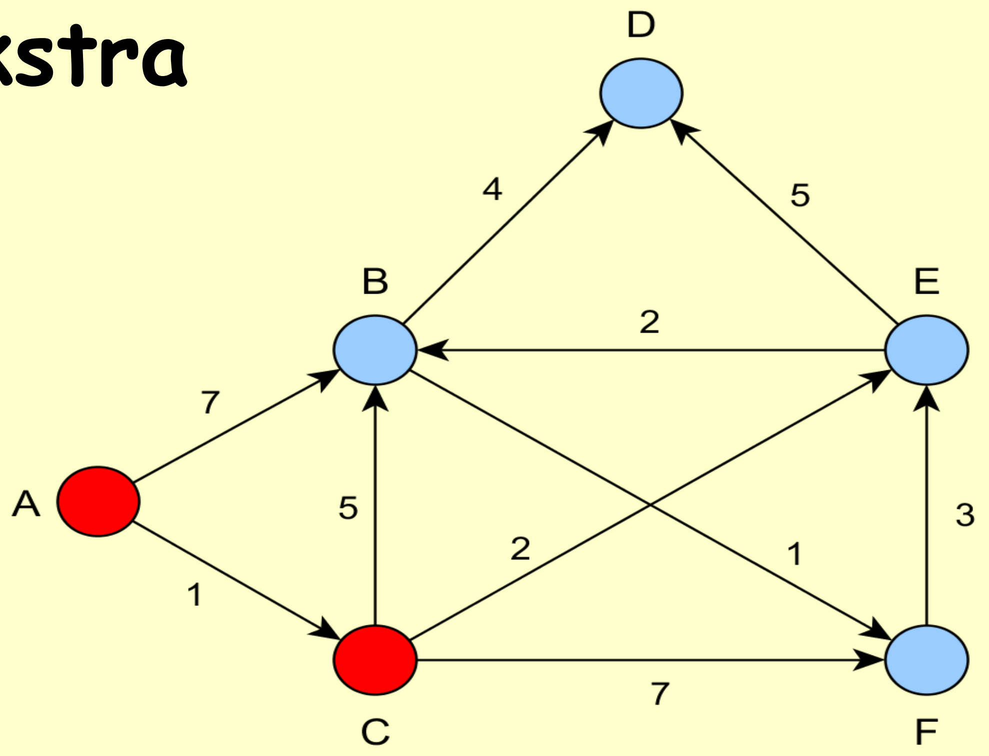
- Exemple



M	$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$
A	0	7	1	$+\infty$	$+\infty$	$+\infty$	A	A	A	$-$	$-$	$-$

Algorithme de Dijkstra

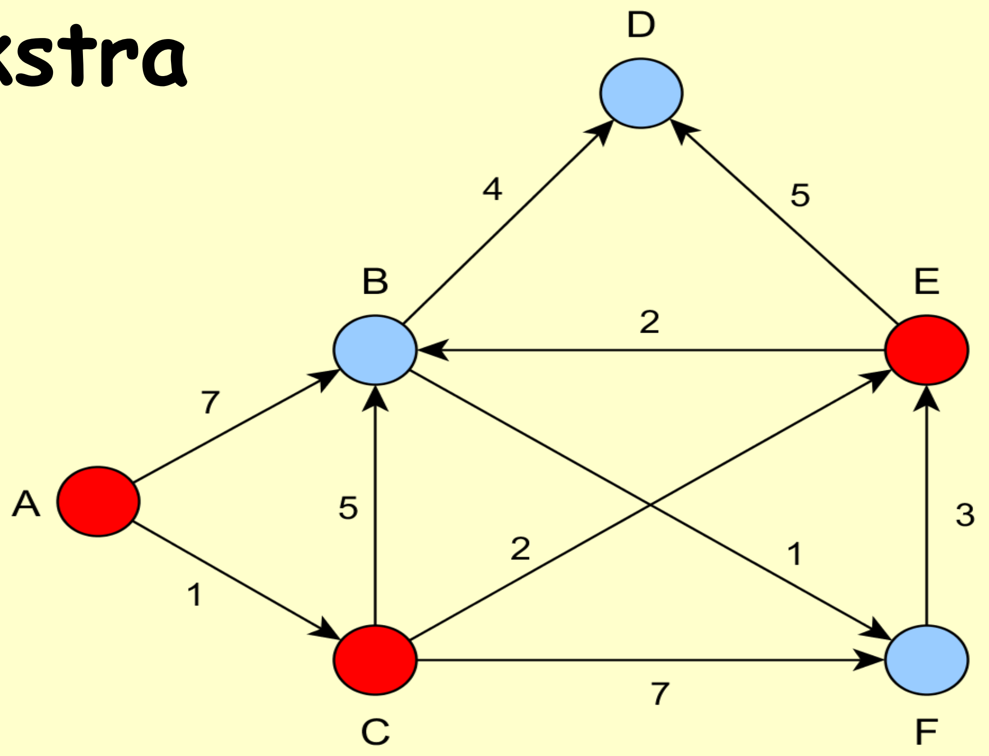
- Exemple



M	$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$
A	0	7	1	$+\infty$	$+\infty$	$+\infty$	A	A	A	$-$	$-$	$-$
A, C	0	6	1	$+\infty$	3	8	A	C	A	$-$	C	C

Algorithme de Dijkstra

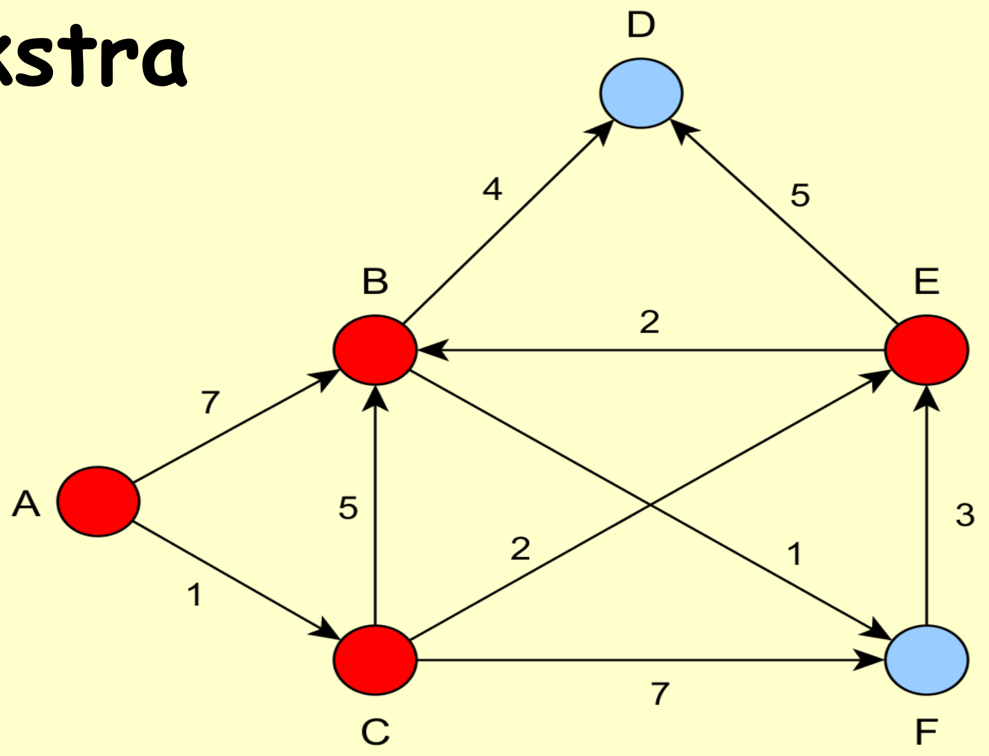
- Exemple



M	$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$
A	0	7	1	$+\infty$	$+\infty$	$+\infty$	A	A	A	$-$	$-$	$-$
A, C	0	6	1	$+\infty$	3	8	A	C	A	$-$	C	C
A, C, E	0	5	1	8	3	8	A	E	A	E	C	C

Algorithme de Dijkstra

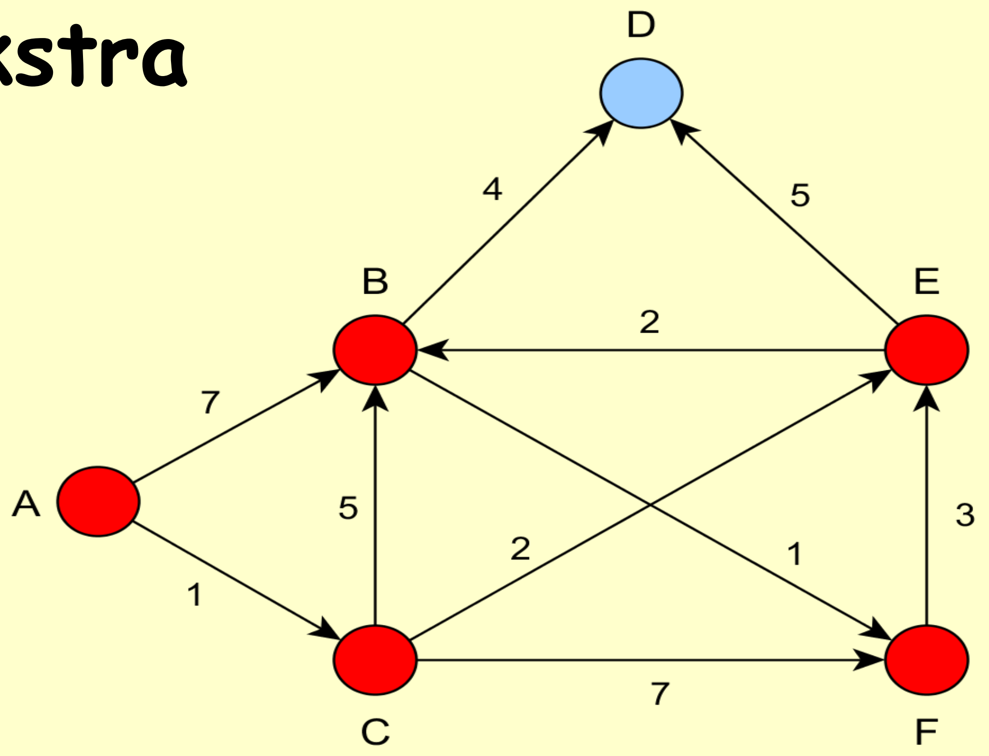
- Exemple



M	$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$
A	0	7	1	$+\infty$	$+\infty$	$+\infty$	A	A	A	$-$	$-$	$-$
A,C	0	6	1	$+\infty$	3	8	A	C	A	$-$	C	C
A,C,E	0	5	1	8	3	8	A	E	A	E	C	C
A,C,E,B	0	5	1	8	3	6	A	E	A	E	C	B

Algorithme de Dijkstra

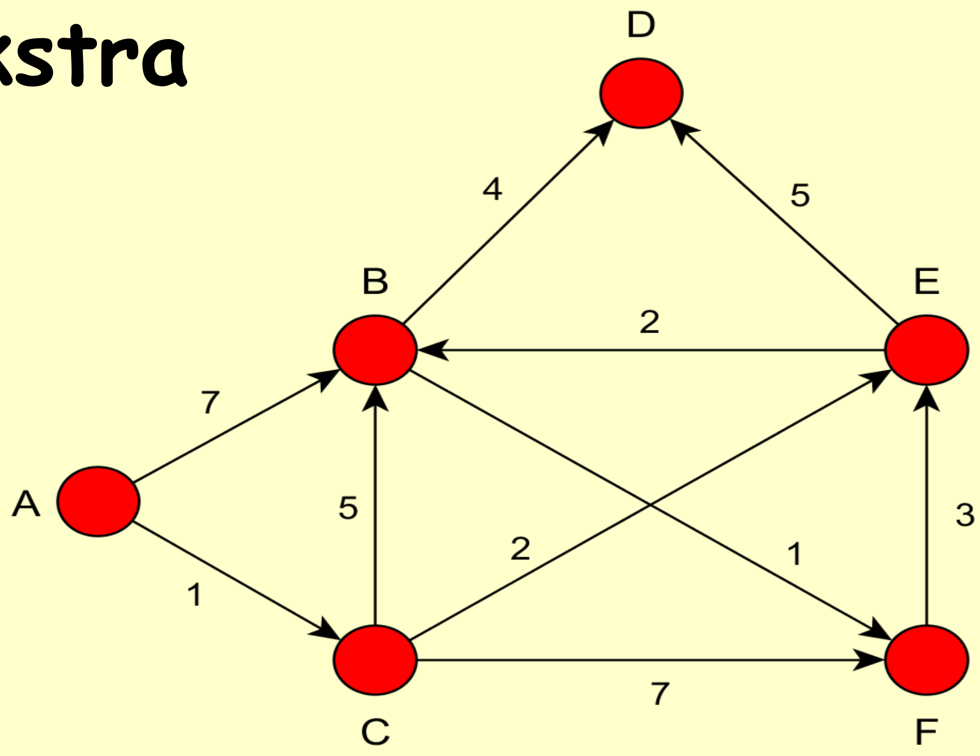
- Exemple



M	$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$
A	0	7	1	$+\infty$	$+\infty$	$+\infty$	A	A	A	$-$	$-$	$-$
A, C	0	6	1	$+\infty$	3	8	A	C	A	$-$	C	C
A, C, E	0	5	1	8	3	8	A	E	A	E	C	C
A, C, E, B	0	5	1	8	3	6	A	E	A	E	C	B
A, C, E, B, F	0	5	1	8	3	6	A	E	A	E	C	B

Algorithme de Dijkstra

- Exemple



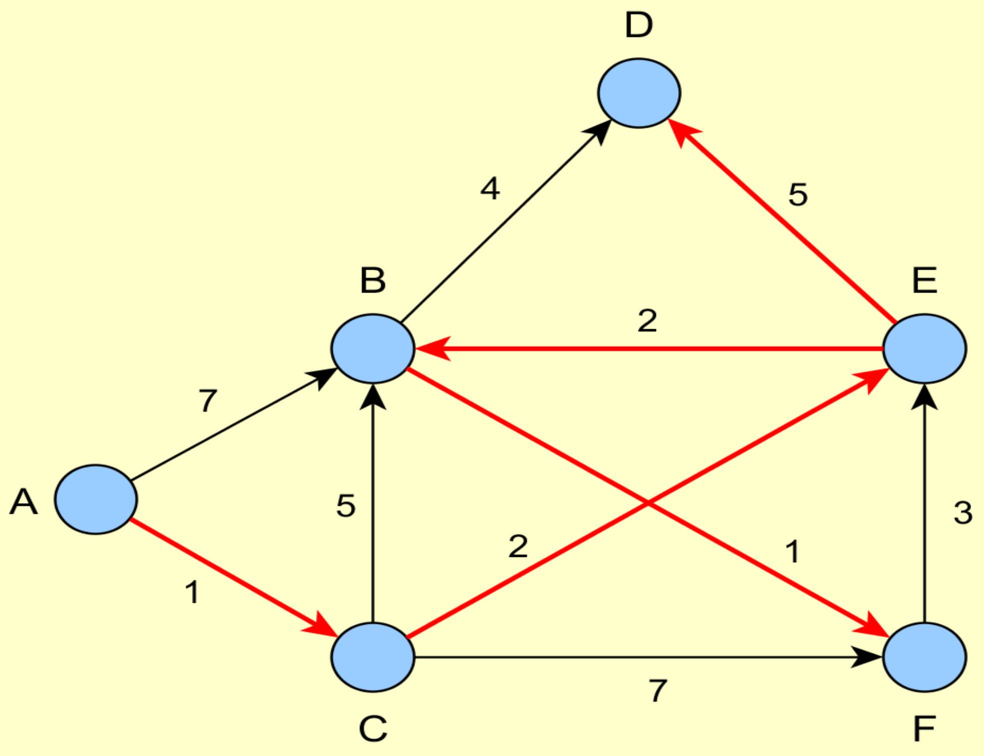
M	$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$
A	0	7	1	$+\infty$	$+\infty$	$+\infty$	A	A	A	–	–	–
A, C	0	6	1	$+\infty$	3	8	A	C	A	–	C	C
A, C, E	0	5	1	8	3	8	A	E	A	E	C	C
A, C, E, B	0	5	1	8	3	6	A	E	A	E	C	B
A, C, E, B, F	0	5	1	8	3	6	A	E	A	E	C	B
A, C, E, B, F, D	0	5	1	8	3	6	A	E	A	E	C	B

Algorithme de Dijkstra

- Le plus court chemin entre A et F en partant de son extrémité (tableau P):
  - Le prédécesseur de F est B.
  - Le prédécesseur de B est E.
  - Le prédécesseur de E est C.
  - Le prédécesseur de C est A.
- Le plus court chemin entre les sommets A et F est donc (A,C,E,B,F).

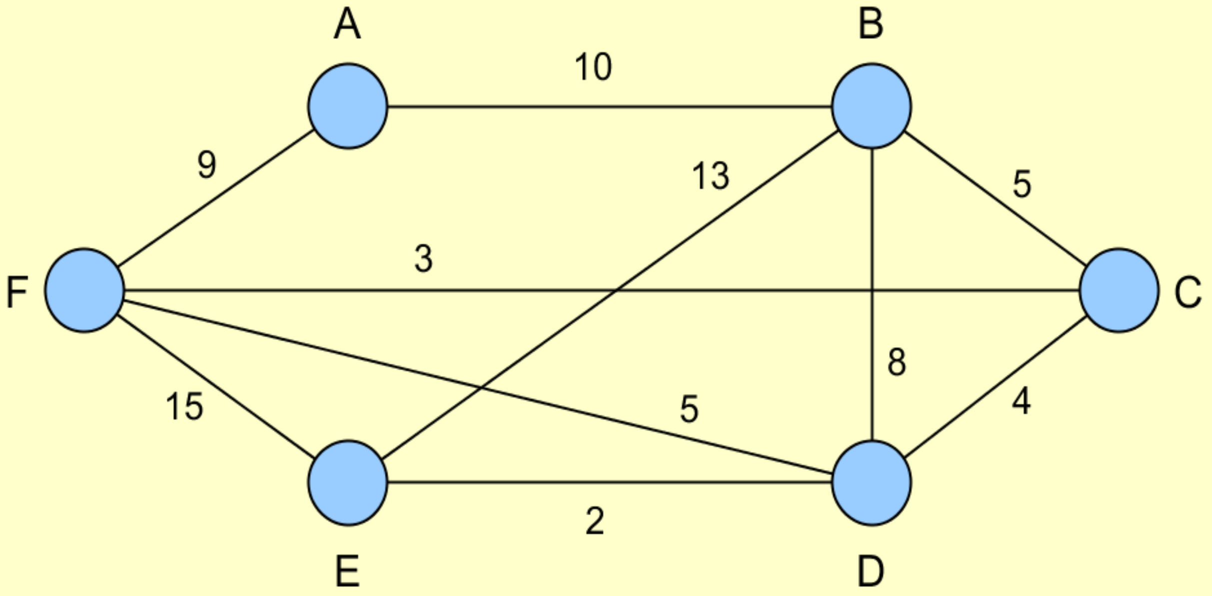
Algorithme de Dijkstra

- Sur ce graphe figurent en rouge tous les plus courts chemins entre le sommet *A* et les autres sommets



Algorithme de Dijkstra

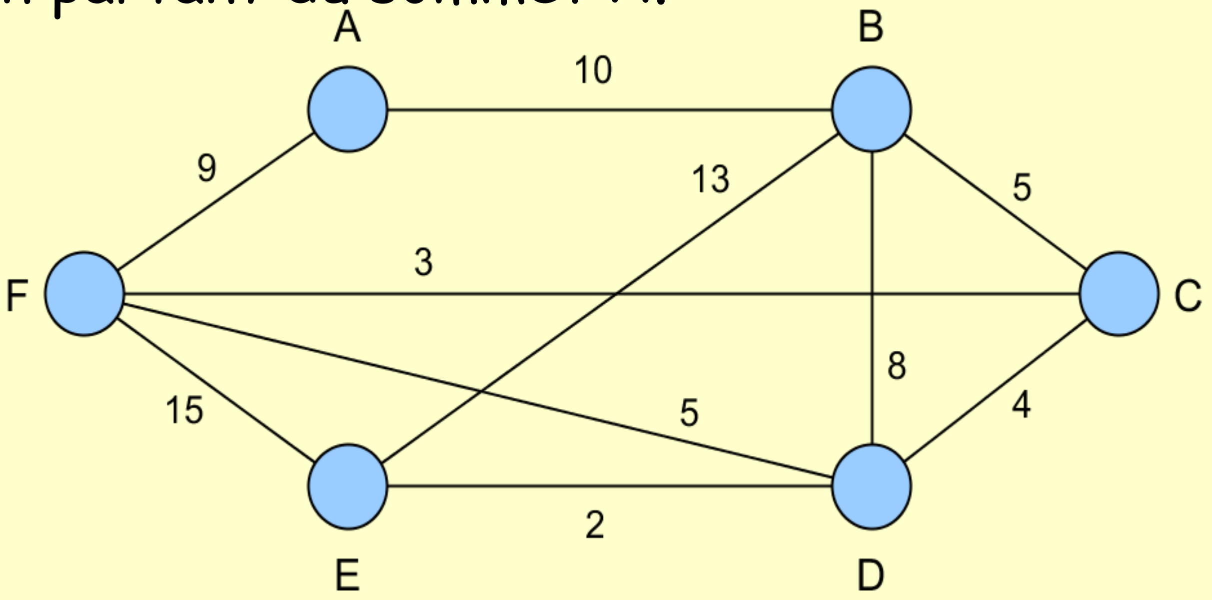
- Exercice :
  - Considérons le graphe non orienté valué suivant :



  - Appliquer l'algorithme de Dijkstra à ce graphe en partant du sommet A.

Algorithme de Dijkstra

- Exercice :
  - Considérons le graphe non orienté valué suivant : Appliquer l'algorithme de Dijkstra à ce graphe en partant du sommet A.

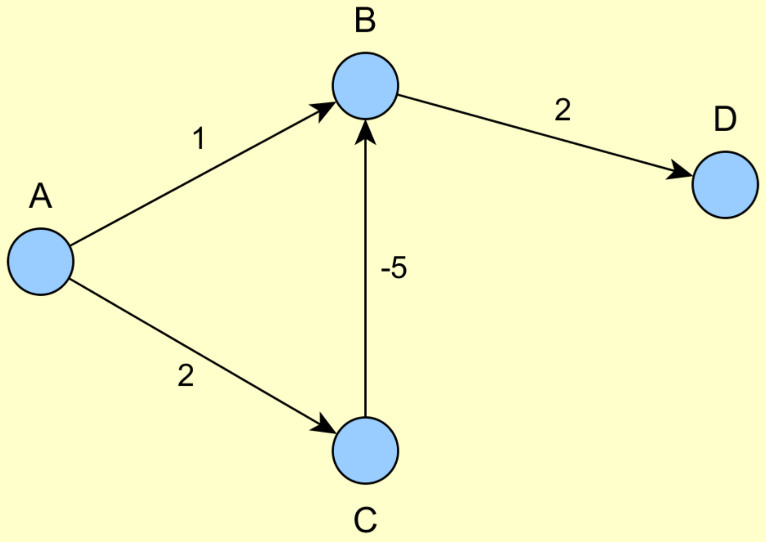


  - Solution :

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$
0	10	12	14	16	9	A	A	F	F	D	A

Algorithme de Dijkstra

- Exercice 2:
  - Considérons le graphe orienté valué suivant :

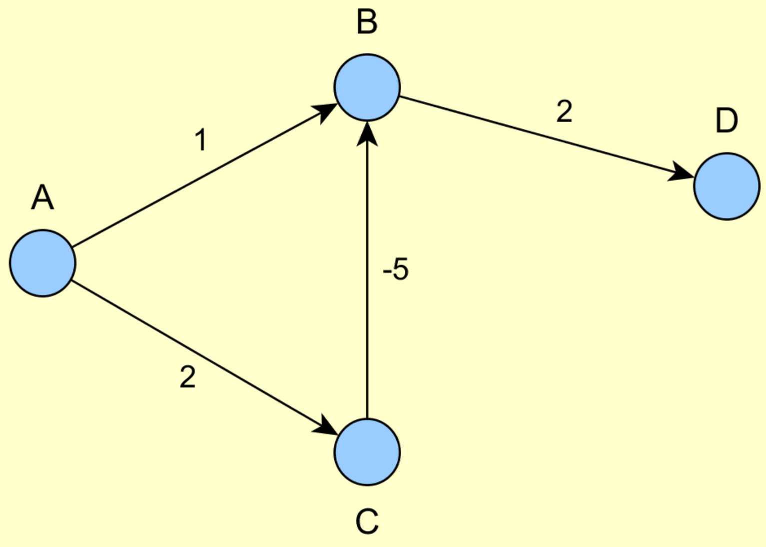


1. Appliquer l'algorithme de Dijkstra à ce graphe en partant du sommet A.
2. Est ce que la distance entre A et D calculée par l'algorithme de Dijkstra est optimale ?
3. Proposer une explication.

Algorithme de Dijkstra

- Exercice 2: **Correction**

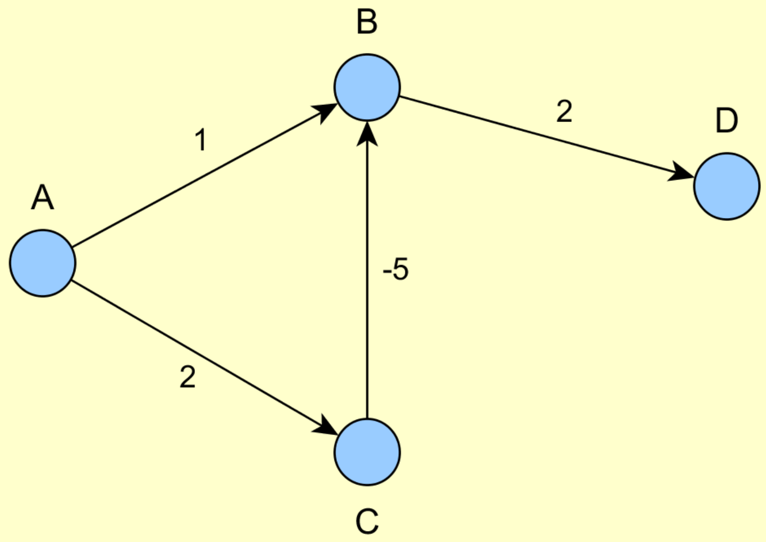
$L[A]$	$L[B]$	$L[C]$	$L[D]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$
0	-3	2	3	A	C	A	B



1. la valeur du chemin (A,C,B,D) retournée par l'algorithme de Dijkstra = 3

2. Ce chemin a une valuation égale à -1, ce qui est mieux que 3.

Algorithme de Dijkstra



- Exercice 2: **Correction**

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$
0	-3	2	3	A	C	A	B

3. Explication:
    - L'ordre dans lequel les sommets ont été traités est A,B,C,D.
    - Lors du traitement du sommet B, la valeur L[D] a été mise à jour et elle n'a pas évolué ensuite car B est le seul prédécesseur de D.
    - Mais, lors du traitement du sommet C, postérieur à celui de B, la valeur négative de la valuation v(C,B) a provoqué une màj de L[B].
    - L'algorithme de Dijkstra ne traitant qu'une seule fois chacun des sommets, cette mise à jour de L[B] n'a pu entraîner celle de L[D].
    - Il "manque" donc des itérations à l'algorithme de Dijkstra en cas de valuations négatives.

Algorithme de Bellman-Ford

Algorithme de Bellman-Ford

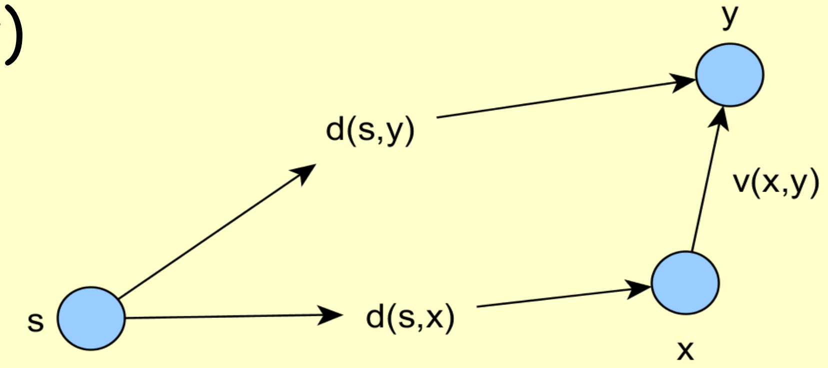
- L'algorithme de Bellman-Ford est un algorithme de recherche de plus court chemin entre un sommet fixé **s** et tous les autres sommets d'un graphe à valuations **quelconques**.

- L'algorithme consiste en deux phases, une d'initialisation la deuxième de traitement (itérations)

- Initialisation
  - On va attribuer la valeur $+\infty$ comme distance provisoire à tous les sommets différents de l'origine.

Algorithme de Bellman-Ford

- Traitement:
  - Lors de chaque itération on va traiter tous les sommets.
  - Cela impliquera qu'à la fin du déroulement de l'algorithme, chaque sommet pourra avoir être traité plusieurs fois.
  - Pour chacun des successeurs y de x, on compare ensuite l'évaluation actuelle de la distance d(s , y) avec la distance d(s , x) + v(x , y)
  - Quand tous les successeurs du sommet x ont été examinés, on passe à un autre sommet mais toujours dans la même itération.



  - Ce qui est la <u>différence majeure</u> avec l'algorithme de Dijkstra

Algorithme de Bellman-Ford

- Soit G = (V , E) un graphe orienté ou non à valuations quelconques d'ordre n.
- On suppose que G ne contient pas de circuits absorbants.
- Soit s : le sommet à partir duquel on va rechercher les plus courts chemins.
- Soit L : un tableau de n cases destiné à contenir les distances de s aux autres sommets.
- Soit P : un tableau de n cases destiné à contenir le prédécesseur de chacun des sommets dans un plus court chemin d'origine s.
- Soit Chnge : un booléen indiquant si l'on doit continuer ou non les itérations.

Algorithme de Bellman-Ford

Initialisation:

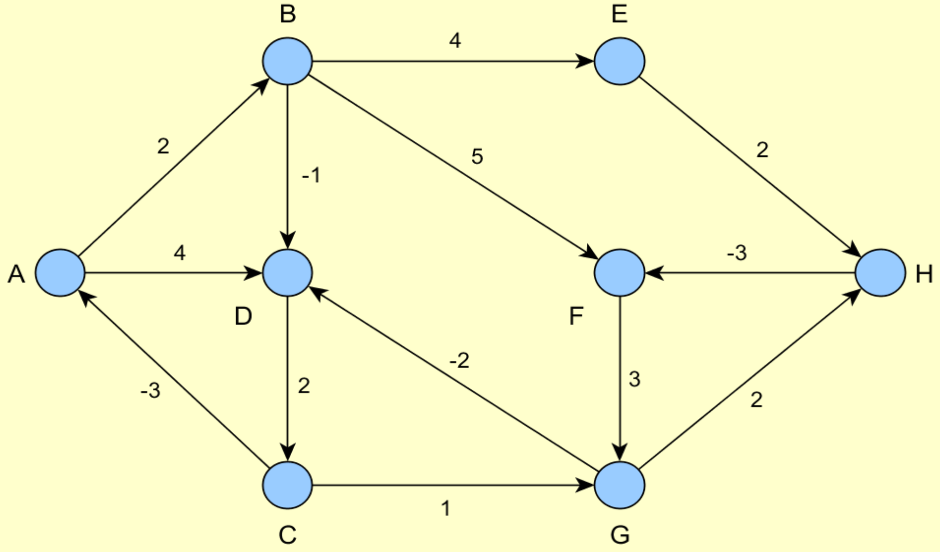
```
L[s] ← 0
P[s] ← s
Pour tout x ≠ s
   L[x] ← +∞
   P[x] ← Ø
FinPour
Chnge ← Vrai
```

Traitement:

```
TantQue Chnge Faire
Chnge ← Faux
Pour tout x ∈ V
    Pour tout y ∈ V tq y soit un successeur de x
        Si L[y] > L[x] + v(x , y) alors
            L[y] > L[x] + v(x , y)
            P[y ] ← x
            Chnge ← Vrai
        FinSi
    FinPour
FinPour
FinTantQue
```

Algorithme de Bellman-Ford

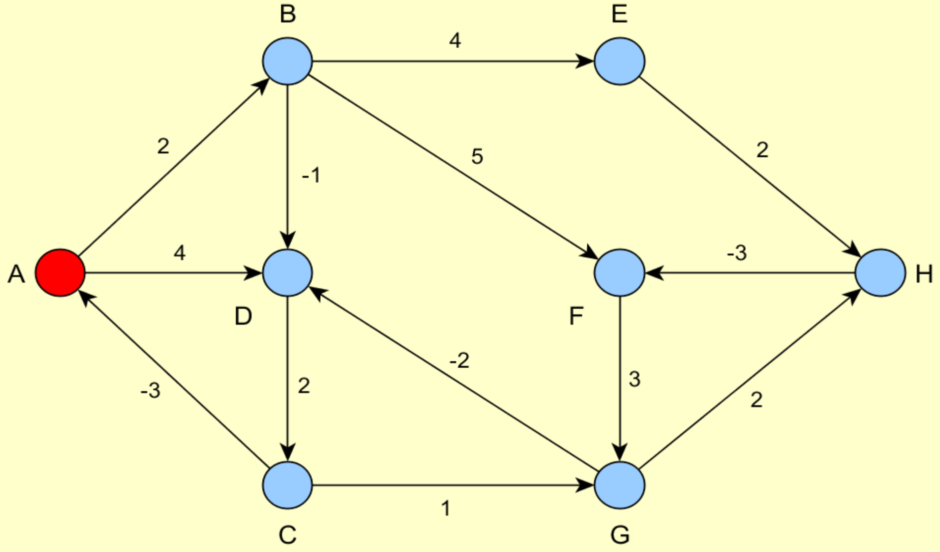
- Exemple:



$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	$-$	$-$	$-$	$-$	$-$	$-$	$-$

Algorithme de Bellman-Ford

- Exemple : Sommet A

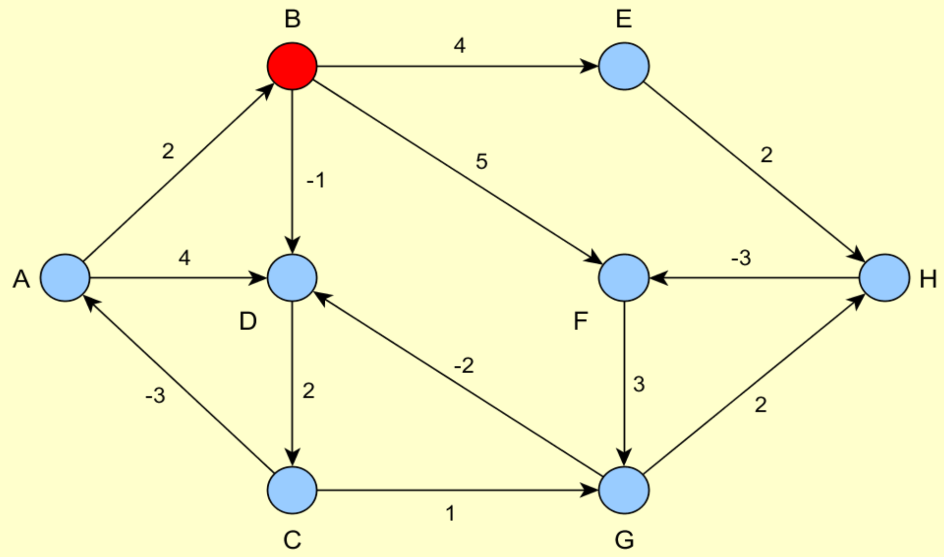


$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–

- On étudie ici les sommets successeurs de A, à savoir B et D.
- La distance courante L[B] du sommet B est égale à +∞.
- On la compare donc avec L[A]+v(A,B), quantité égale à 0 + 2 = 2.
- Il faut màj la distance L[B], ainsi que le prédécesseur de B qui devient A.
- Même chose pour le sommet D.

Algorithme de Bellman-Ford

- Exemple: Sommet B

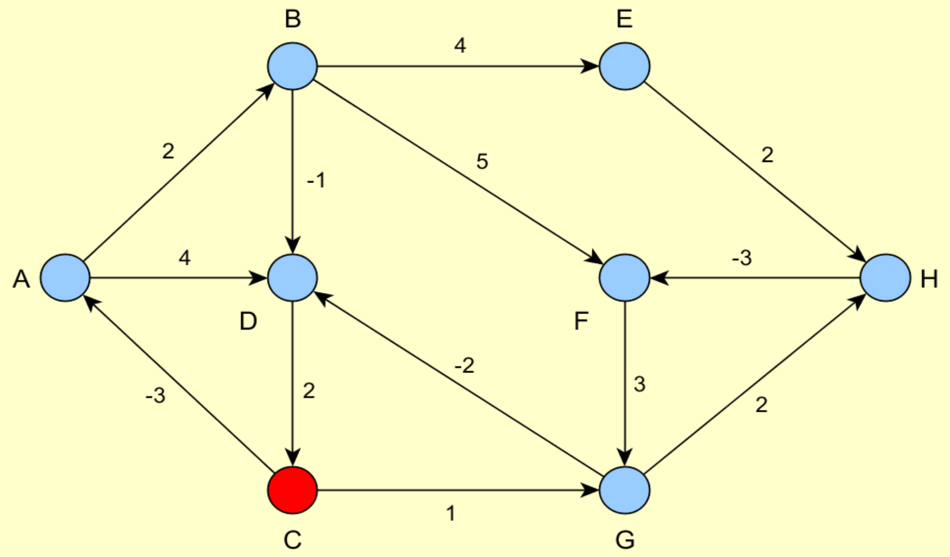


$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–

- On étudie ici les sommets successeurs de B, à savoir D, E et F.
- La distance courante L[D] du sommet D est égale à 4.
- On la compare donc avec L[B]+v(B,D), quantité égale à 2+(-1)=1.
- Il faut màj la distance L[D], ainsi que le prédécesseur de D qui devient B.
- Même chose pour les sommet E et F.

Algorithme de Bellman-Ford

- Exemple: Sommet C

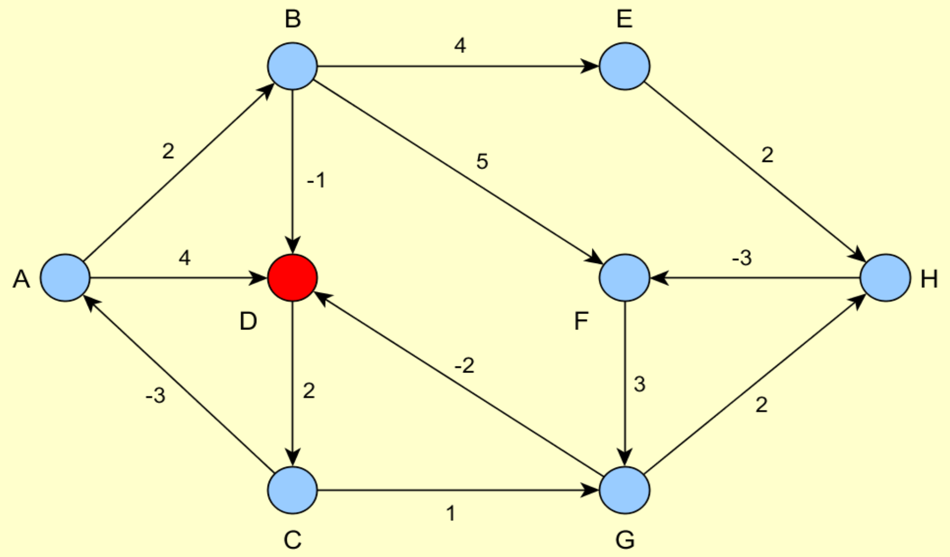


$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–

- Puisque la distance courante L[C] du sommet C est égale à +∞, il n'y aura aucune màj à faire sur ses successeurs.

Algorithme de Bellman-Ford

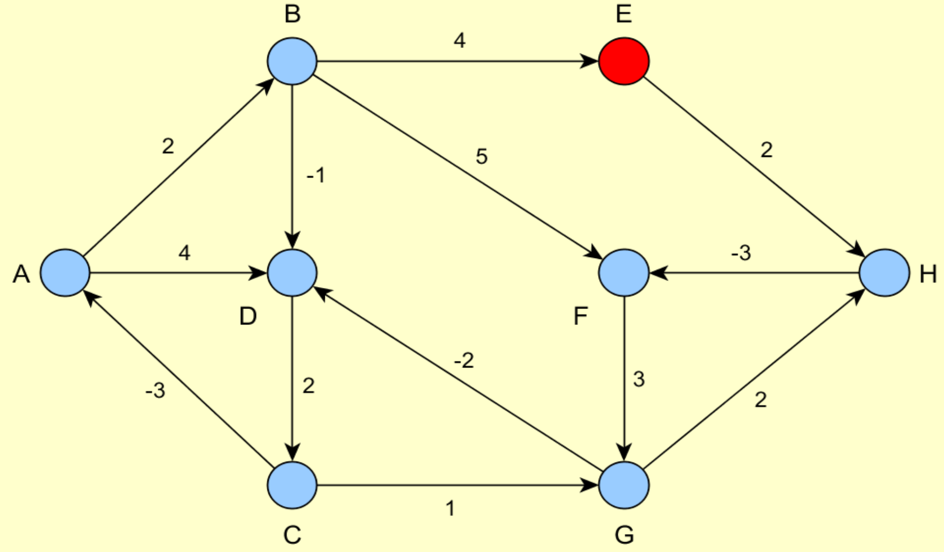
- Exemple: sommet D



$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–

- On étudie ici l'unique sommet successeur de D, à savoir C.
- La distance courante L[C] du sommet C est égale à +∞.
- On la compare donc avec L[D]+v(D,C), quantité égale à 1+2=3.
- Il faut donc mettre à jour la distance L[C], ainsi que le prédécesseur de C qui devient D.

Algorithme de Bellman-Ford

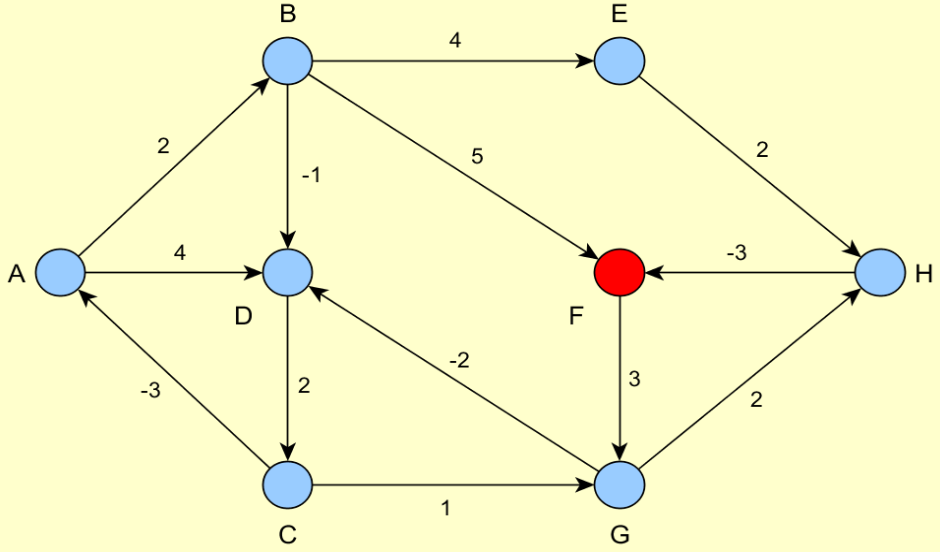


- Exemple: sommet E

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	–	E

- étude de l'unique sommet successeur de E, à savoir H.
- La distance courante L[H] du sommet H est égale à +∞.
- On la compare donc avec L[E]+v(E,H), quantité égale à 6+2=8.
- Il faut donc mettre à jour la distance L[H], ainsi que le prédécesseur de H qui devient E.

Algorithme de Bellman-Ford

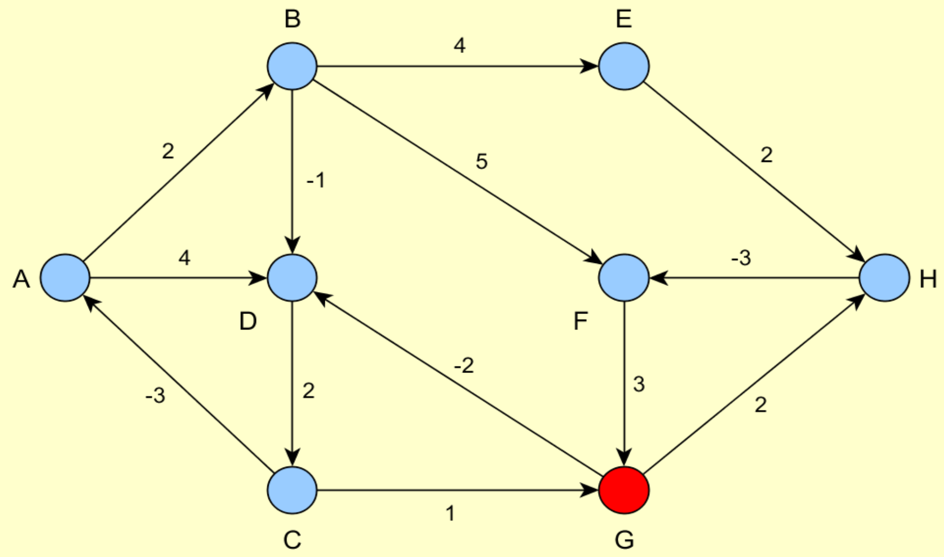


- Exemple: sommet F

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	–	E
0	2	3	1	6	7	10	8	A	A	D	B	B	B	F	E

- La distance courante L[G] du sommet G est égale à +∞
- On la compare donc avec L[F]+v(F,G), quantité égale à 7 + 3 = 10
- Il faut màj la distance L[G], ainsi que le prédécesseur de G qui devient F.

Algorithme de Bellman-Ford



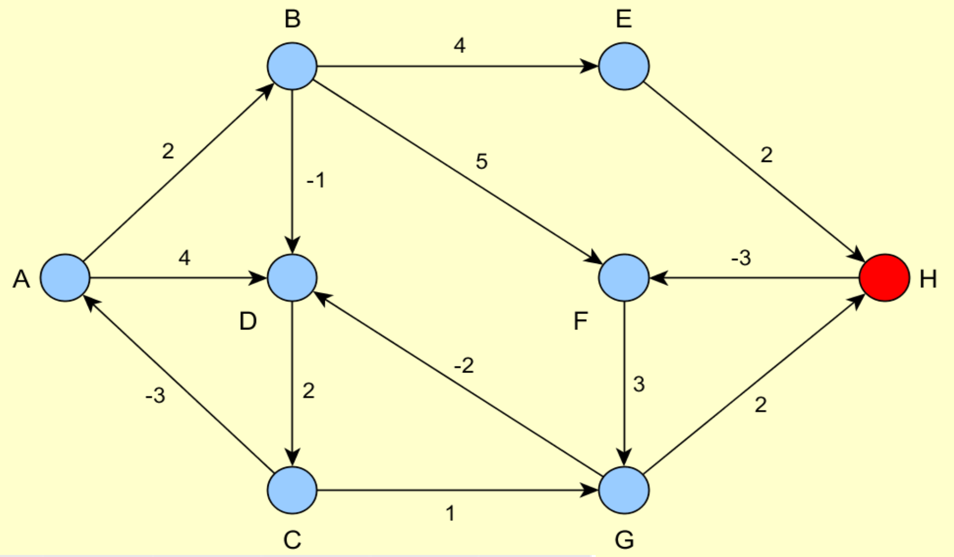
- Exemple: sommet G

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	–	E
0	2	3	1	6	7	10	8	A	A	D	B	B	B	F	E

- La distance courante L[D] du sommet D est égale à 1. On la compare avec L[G]+v(G,D), quantité égale à 10+(-2)=8

- Il ne faut pas màj la distance L[D] et P[D] non plus, de même pour le sommet H.

Algorithme de Bellman-Ford



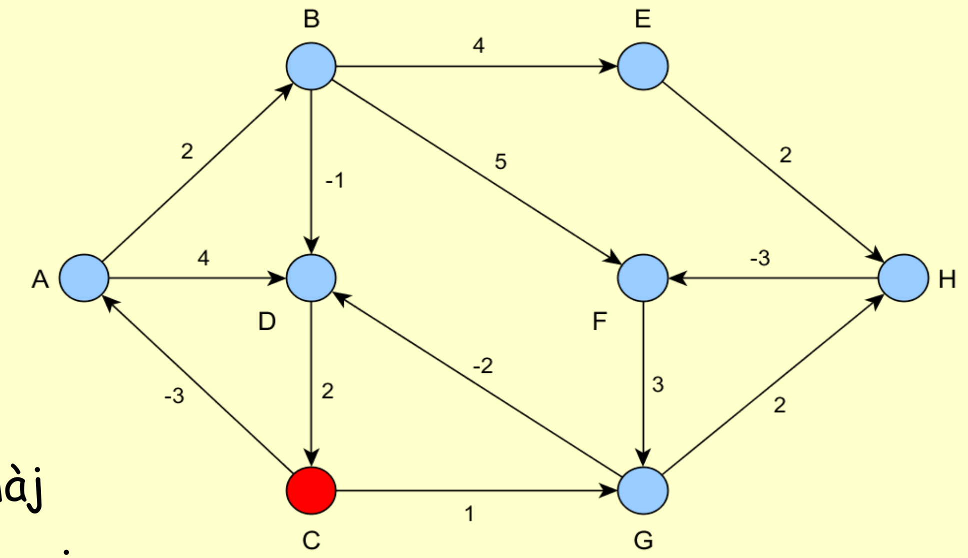
- Exemple: sommet H

- On étudie le sommet successeur de H, à savoir F.
- La distance courante L[F] du sommet F est égale à 7.

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	–	E
0	2	3	1	6	7	10	8	A	A	D	B	B	B	F	E
0	2	3	1	6	5	10	8	A	A	D	B	B	H	F	E

- On la compare donc avec L[H]+v(H,F), quantité égale à 8+(-3)=5.
- Il faut màj la distance L[F], ainsi que le prédécesseur de F qui devient H.

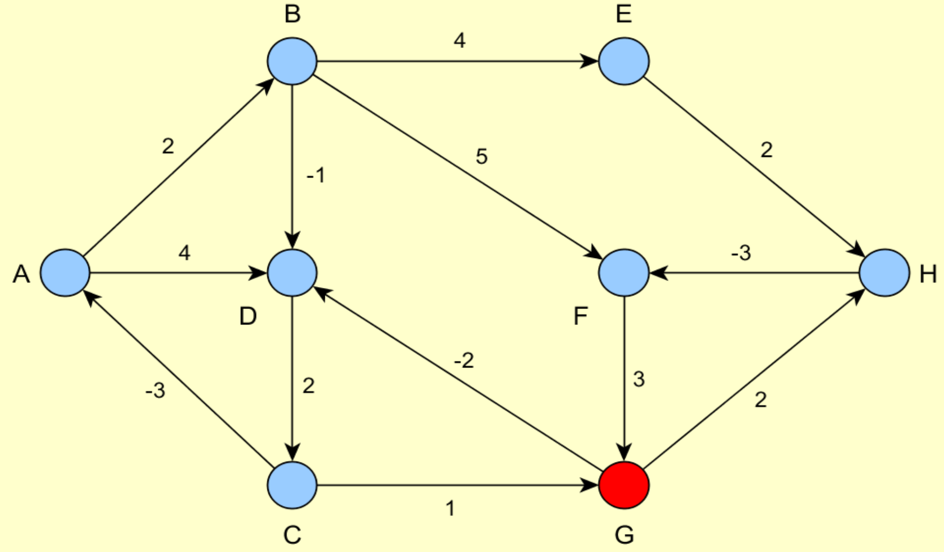
Algorithme de Bellman-Ford



- Exemple:
  - La première itération est terminée.
  - Lors de la seconde, Re-traiter tous les sommets un par un.
  - Nous n'indiquerons ici que les sommets qui ont conduit à une màj des distances mais il est cependant nécessaire de tous les examiner.

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	–	E
0	2	3	1	6	7	10	8	A	A	D	B	B	B	F	E
0	2	3	1	6	5	10	8	A	A	D	B	B	H	F	E
0	2	3	1	6	5	4	8	A	A	D	B	B	H	C	E

Algorithme de Bellman-Ford

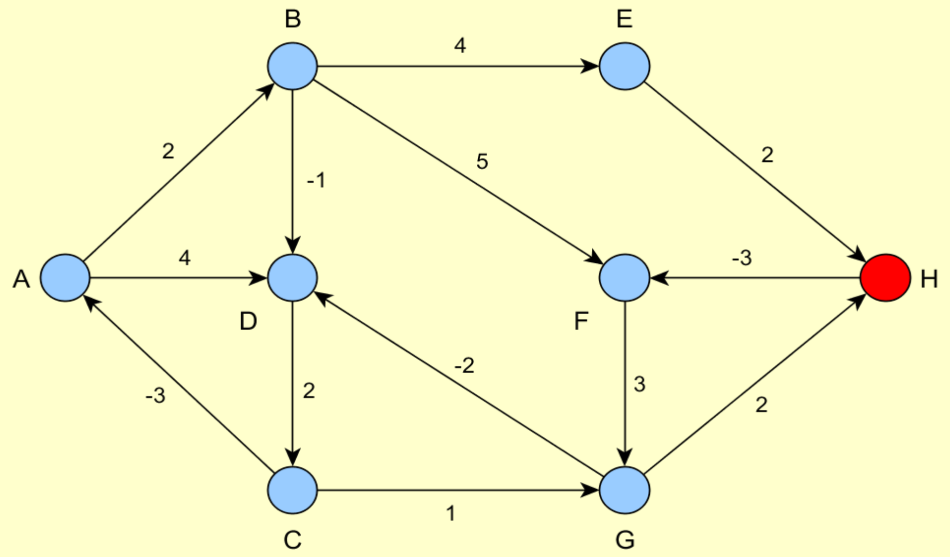


- Exemple:
  - Itération 2

$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	−	−	−	−	−	−	−
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	−	A	−	−	−	−
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	−	B	B	B	−	−
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	−	−
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	−	E
0	2	3	1	6	7	10	8	A	A	D	B	B	B	F	E
0	2	3	1	6	5	10	8	A	A	D	B	B	H	F	E
0	2	3	1	6	5	4	8	A	A	D	B	B	H	C	E
0	2	3	1	6	5	4	6	A	A	D	B	B	H	C	G

Algorithme de Bellman-Ford

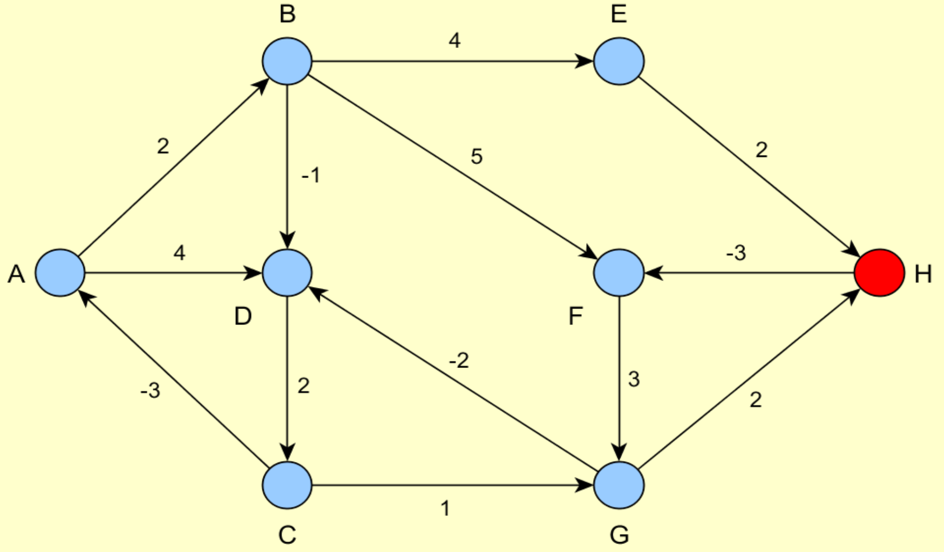
- Exemple:
  - Itération 3



$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	–	E
0	2	3	1	6	7	10	8	A	A	D	B	B	B	F	E
0	2	3	1	6	5	10	8	A	A	D	B	B	H	F	E
0	2	3	1	6	5	4	8	A	A	D	B	B	H	C	E
0	2	3	1	6	5	4	6	A	A	D	B	B	H	C	G
0	2	3	1	6	3	4	6	A	A	D	B	B	H	C	G

Algorithme de Bellman-Ford

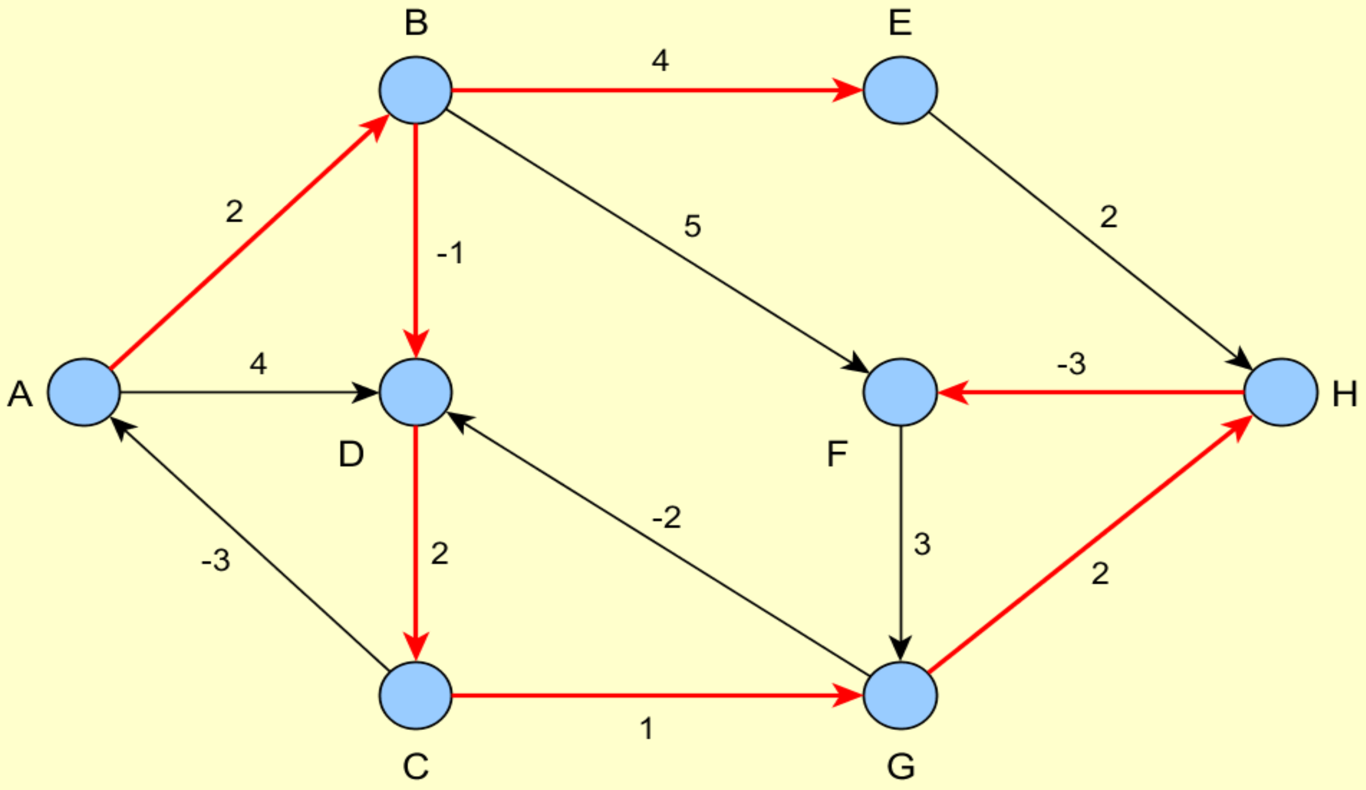
- Exemple:
  - Itération 4



$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$L[F]$	$L[G]$	$L[H]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$	$P[F]$	$P[G]$	$P[H]$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	–	–	–	–	–	–	–
0	2	$+\infty$	4	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A	A	–	A	–	–	–	–
0	2	$+\infty$	1	6	7	$+\infty$	$+\infty$	A	A	–	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	$+\infty$	A	A	D	B	B	B	–	–
0	2	3	1	6	7	$+\infty$	8	A	A	D	B	B	B	–	E
0	2	3	1	6	7	10	8	A	A	D	B	B	B	F	E
0	2	3	1	6	5	10	8	A	A	D	B	B	H	F	E
0	2	3	1	6	5	4	8	A	A	D	B	B	H	C	E
0	2	3	1	6	5	4	6	A	A	D	B	B	H	C	G
0	2	3	1	6	3	4	6	A	A	D	B	B	H	C	G

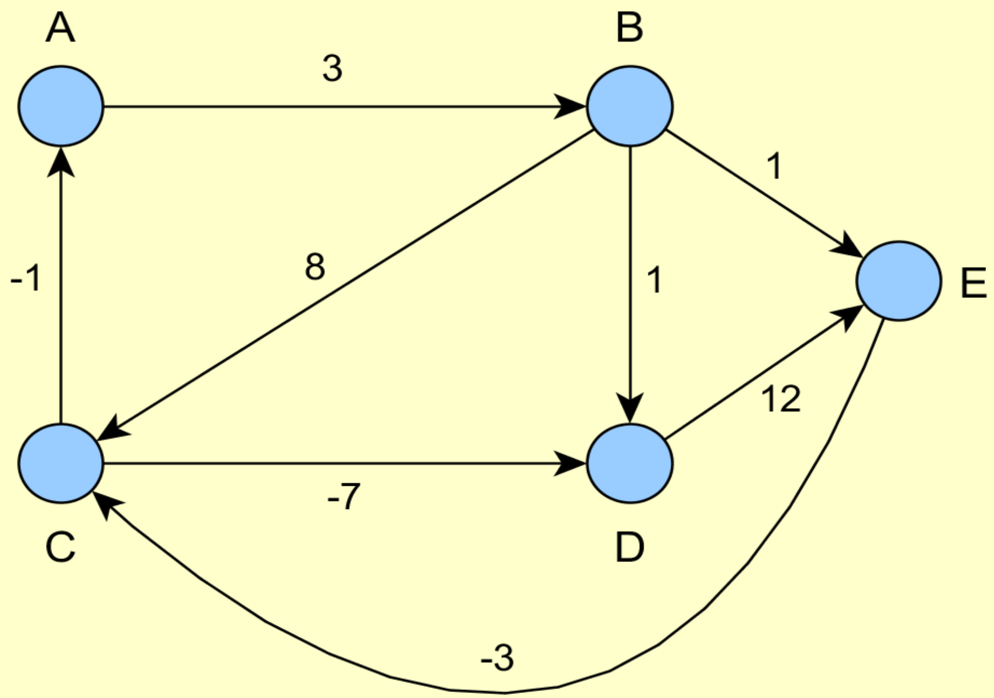
Algorithme de Bellman-Ford

- Sur le graphe ci-contre, figurent en rouge tous les plus courts chemins entre le sommet A et les autres sommets



Algorithme de Bellman-Ford

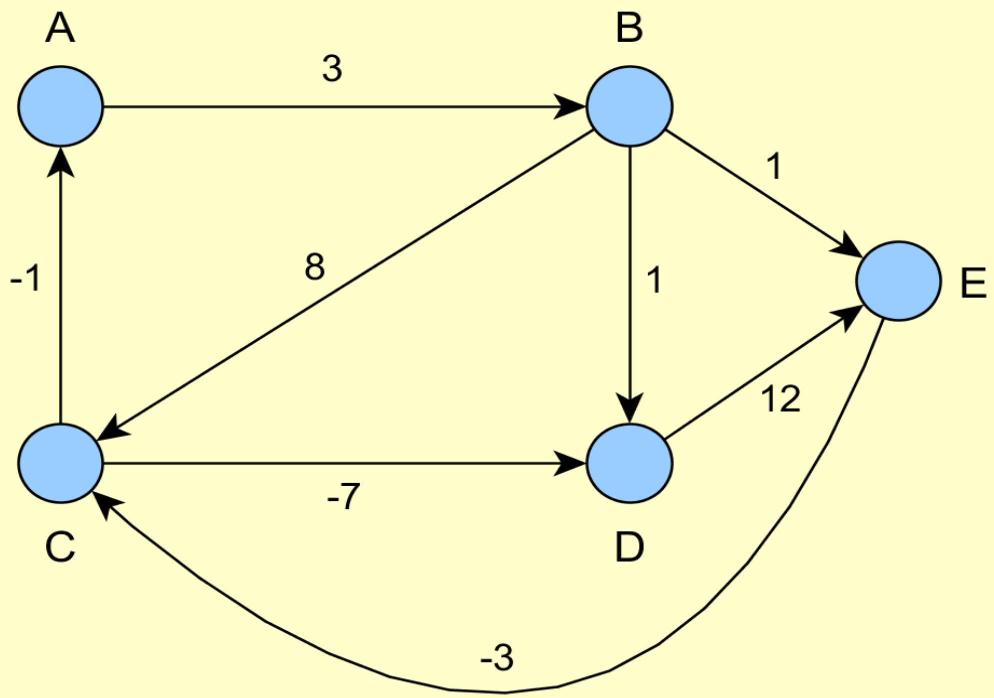
- Exercice :
  - Considérons le graphe orienté valué suivant :



  - Appliquer l'algorithme de Bellman-Ford à ce graphe en partant du sommet A.

Algorithme de Bellman-Ford

- Exercice : Solution
  - Considérons le graphe orienté valué suivant :



$L[A]$	$L[B]$	$L[C]$	$L[D]$	$L[E]$	$P[A]$	$P[B]$	$P[C]$	$P[D]$	$P[E]$
0	3	1	-6	4	A	A	E	C	B

TD 4

- Dans le graphe orienté $G = (X, U)$ ci-contre, valué par des longueurs d'arcs positives :
  - Utiliser l'algorithme de Dijkstra, pour déterminer le plus cours chemin depuis le sommet a jusqu'au sommet f.
  - Refaire la question via l'algorithme de Bellman-Ford