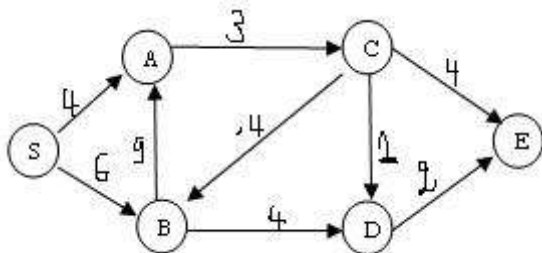


Serie n°3

Exercise 1. Let the graph below:



1. Find the shortest path from S to all other vertices by applying:

a. Dijkstra's algorithm:

ALGORITHME_DJKSTRA(cG, s)

Entrée $G=(X,E,V)$: graphe orienté ou non avec V l'ensemble des poids des arcs ($\in \mathbb{R}^+$), s : sommet de G %
sommet de départ

Sortie distance : un tableau des distances entre s et les autres sommets

Début

% initialisation initialiser le tableau *distance* de taille $|X|$

distance[i] $\leftarrow +\infty, \forall i \in X - \{s\}$;

distance[s] $\leftarrow 0$;

% itérations

Pour tout it allant de 1 à $|X| - 1$ faire

Sélectionner le sommet i ayant la plus petite valeur de distance et marquer le comme terminé ;

Pour tout sommet j adjacent à i non encore sélectionné faire

Si distance[i] + poids(i,j) < distance[j] alors

distance[j] $\leftarrow$ distance[i] + poids(i,j);

% on écrit également dans la même case j de quel sommet vient j (sommet i)

Sinon

recopier ce qui se trouvait dans la ligne précédente

Compléter la ligne du tableau en recopiant les valeurs de la ligne précédente.

Fin si

Fin pour

Fin pour

retourner distance

Fin

Solution :

1. Au début, on initialise les distances par $+\infty$ pour tous les sommets sauf le sommet de départ par un 0, et on fixe le nombre des itérations par $|X|-1=6-1=5$

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1						
2						
3						
4						
5						

Pour remplir le tableau, on commence de calculer la distance le sommet S et les autres sommets : on commence par les sommets adjacents :

$i = S$, on prend j le sommet A, on essaie d'utiliser cette instruction de test pour vérifier la distance et de remplir le tableau.

Si distance[i] + poids(i,j) < distance[j] **alors**

$\text{distance}[S] + \text{poids}(S,A) = 0+4=4$; $\text{distance}[A]= +\infty \rightarrow 4 < +\infty$ donc:

$\text{distance}[A]= \text{distance}[S] + \text{poids}(S,A) = 4$; on remplit la case de sommet A par la distance calculée et on met de quel sommet vient de cette façon:

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1		4 - S				
2						
3						
4						
5						

On met la même chose avec les autres sommets : i = S, on prend j le sommet B, on essaie d'utiliser cette instruction de test pour vérifier la distance et de remplir le tableau.

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[S] + \text{poids}(S,B) = 0+6=6$; $\text{distance}[B]= +\infty \rightarrow 6 < +\infty$, donc:

$\text{distance}[B]= \text{distance}[S] + \text{poids}(S,B) = 6$; on remplit la case de sommet B par la distance calculée et on met de quel sommet vient et on met les mêmes valeurs précédents $+\infty$ pour les distances des autres sommets car sont pas adjacents avec le sommet s :

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1		4 - S	6 - S	$+\infty$	$+\infty$	$+\infty$
2						
3						
4						
5						

La deuxième itération, on considère le sommet S visité. On choisit la distance la plus petite ici est 4 de S à A. Donc on prend le sommet A et on calcule la distance de sommet adjacent de A : C.

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[A] + \text{poids}(A,C) = 4+3=7$; $\text{distance}[C]= +\infty \rightarrow 7 < +\infty$, donc:

$\text{distance}[C]= \text{distance}[A] + \text{poids}(A,C) = 7$; on remplit la case de sommet C par la distance calculée et on met de quel sommet vient de cette façon:

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1		4 - S	6 - S	$+\infty$	$+\infty$	$+\infty$
2			6 - S	7 - A		
3						
4						
5						

Le sommet adjacent de B est D , on essaie d'utiliser cette instruction de test pour vérifier la distance et de remplir le tableau.

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[B] + \text{poids}(B,D) = 6 + 4=10$; $\text{distance}[D]= +\infty \rightarrow 10 < +\infty$, donc:

$\text{distance}[D]= \text{distance}[B] + \text{poids}(B,D) = 10$; on remplit la case de sommet D par la distance calculée et on met de quel sommet vient et on met les mêmes valeurs $+\infty$ pour les distances des autres sommets car sont pas adjacents avec le sommet s :

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1		4 - S	6 - S	$+\infty$	$+\infty$	$+\infty$
2			6 - S	7 - A	$+\infty$	$+\infty$
3				7 - A	10 - B	$+\infty$
4						
5						

Le sommet adjacent de C est D, E. On essaie d'utiliser cette instruction de test pour vérifier la distance et de remplir le tableau.

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[C] + \text{poids}(C,D) = 7 + 1=8$; $\text{distance}[D]= +\infty \rightarrow 8 < +\infty$, donc:

$\text{distance}[D] = \text{distance}[C] + \text{poids}(C,D) = 8$; on remplit la case de sommet D par la distance calculée et on met de quel sommet vient ;

$\text{distance}[C] + \text{poids}(C,E) = 7 + 4=11$; $\text{distance}[E]= +\infty \rightarrow 11 < +\infty$, donc:

$\text{distance}[E] = \text{distance}[C] + \text{poids}(C,E) = 11$; on remplit la case de sommet E par la distance calculée et on met de quel sommet vient ;

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1		4 - S	6 - S	$+\infty$	$+\infty$	$+\infty$
2			6 - S	7 - A	$+\infty$	$+\infty$
3				7 - A	10 - B	$+\infty$
4					8 - C	11 - E
5						

Le sommet adjacent de D est E. On essaie d'utiliser cette instruction de test pour vérifier la distance et de remplir le tableau.

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[D] + \text{poids}(D,E) = 8 + 2=10$; $\text{distance}[E]= +\infty \rightarrow 10 < +\infty$, donc:

$\text{distance}[E] = \text{distance}[D] + \text{poids}(D,E) = 10$; on remplit la case de sommet E par la distance calculée et on met de quel sommet vient ;

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1		4 - S	6 - S	$+\infty$	$+\infty$	$+\infty$
2			6 - S	7 - A	$+\infty$	$+\infty$
3				7 - A	10 - B	$+\infty$
4					8 - C	11 - E
5						10 - D

Les plus courts chemins à partir de S sont :

- SA distance =4
- SB distance = 6
- SAC distance = 7

- **SACD distance = 8**
- **SACDE distance = 10**

b. Bellman-Ford's algorithm.

Algorithme de Bellman-Ford

ALGORITHME_BELLMAN_FORD(G, s)
 Entree $G=(X,E,V)$: graphe oriente avec V l'ensemble des poids des arcs ($\in \mathbb{R}$)
 s :sommet de G
 Sortie distance : un tableau des distances entre s et les autres sommets
 predecesseur: un tableau des predecesseurs des sommets
Debut
 % initialisation
 initialiser les tableaux *distance* et *predecesseur* de taille $|X|$
 $distance[i] \leftarrow +\infty, \forall i \in X - \{s\}$;
 $distance[s] \leftarrow 0$;
 $predecesseur[i] \leftarrow nil, \forall i \in X$
 % relaxation
 Pour tout it allant de 1 a $|X| - 1$ faire
 Pour tout arc $(i, j) \in E$ faire
 Si $distance[i] + poids(i,j) < distance[j]$ alors
 $distance[j] \leftarrow distance[i] + poids(i,j)$;
 $predecesseur[j] \leftarrow i$;
 Fin si
 Fin pour
 Fin pour
 % detection de circuits absorbant
 Si $\exists$ arc $(i, j) \in E$ tel que $distance[i] + poids(i,j) < distance[j]$ alors STOP % existence
 d'un circuit absorbant : pas de plus court chemin
 Fin si
 retourner distance et predecesseur Fin

Initialisation :

Ordre d'examen des arcs :

Arc	(S,A)	(S,B)	(A,C)	(B,A)	(B,D)	(C,B)	(C,D)	(C,E)	(D,E)
Poids	4	6	3	9	4	-4	1	4	2

Tableau des distances

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1						
2						
3						

Tableau des predecesseurs

	A	B	C	D	E
init	Nil	Nil	Nil	Nil	Nil
1					
2					
3					

La première itération :

Pour tout arc $(i, j) \in E$ faire

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[j] \leftarrow \text{distance}[i] + \text{poids}(i,j)$;

$\text{predecesseur}[j] \leftarrow i$;

Arc (S,A) :

$\text{distance}[S] + \text{poids}(S,A) = 0 + 4 = 4$, $\text{distance}[j] = \text{distance}[A] = +\infty$; $4 < +\infty$

$\text{distance}[S] + \text{poids}(S,A) < \text{distance}[A]$

$\text{distance}[A] = \text{distance}[S] + \text{poids}(S,A) = 4$

$\text{predecesseur}[A] \leftarrow S$;

Arc (S,B) :

$\text{distance}[S] + \text{poids}(S,B) = 0 + 6 = 6$, $\text{distance}[j] = \text{distance}[B] = +\infty$; $6 < +\infty$

$\text{distance}[S] + \text{poids}(S,B) < \text{distance}[B]$

$\text{distance}[B] = \text{distance}[S] + \text{poids}(S,B) = 6$

$\text{predecesseur}[B] \leftarrow S$;

Arc (A,C) :

$\text{distance}[A] + \text{poids}(A,C) = 4 + 3 = 7$, $\text{distance}[j] = \text{distance}[C] = +\infty$; $7 < +\infty$

$\text{distance}[A] + \text{poids}(A,C) < \text{distance}[C]$

$\text{distance}[C] = \text{distance}[A] + \text{poids}(A,C) = 7$

$\text{predecesseur}[C] \leftarrow A$;

Arc (B,A) :

$\text{distance}[B] + \text{poids}(B,A) = 6 + 9 = 15$, $\text{distance}[j] = \text{distance}[A] = 4$; $15 > 4$

$\text{distance}[A] = 4$, pas de changement.

$\text{predecesseur}[A] \leftarrow S$; reste S.

Arc (B,D) :

$\text{distance}[B] + \text{poids}(B,D) = 6 + 4 = 10$, $\text{distance}[j] = \text{distance}[D] = +\infty$; $10 < +\infty$

$\text{distance}[B] + \text{poids}(B,D) < \text{distance}[D]$

$\text{distance}[D] = \text{distance}[B] + \text{poids}(B,D) = 10$

$\text{predecesseur}[D] \leftarrow B$;

Arc (C,B) :

$\text{distance}[C] + \text{poids}(C,B) = 7 - 4 = 3$, $\text{distance}[j] = \text{distance}[B] = 6$; $3 < 6$

$\text{distance}[C] + \text{poids}(C,B) < \text{distance}[B]$

$\text{distance}[B] = \text{distance}[C] + \text{poids}(C,B) = 3$

$\text{predecesseur}[B] \leftarrow C$;

Arc (C,D) :

$\text{distance}[C] + \text{poids}(C,D) = 7 + 1 = 8$, $\text{distance}[j] = \text{distance}[D] = 10$; $8 < 10$

$\text{distance}[C] + \text{poids}(C,D) < \text{distance}[D]$

$\text{distance}[D] = \text{distance}[C] + \text{poids}(C,D) = 8$

$\text{predecesseur}[D] \leftarrow C$;

Arc (C,E) :

$\text{distance}[C] + \text{poids}(C,E) = 7 + 4 = 11$, $\text{distance}[j] = \text{distance}[E] = +\infty$; $11 < +\infty$

$\text{distance}[C] + \text{poids}(C,E) < \text{distance}[E]$

$\text{distance}[E] = \text{distance}[C] + \text{poids}(C,E) = 11$

$\text{predecesseur}[E] \leftarrow C$;

Arc (D,E) :

$\text{distance}[D] + \text{poids}(D,E) = 8 + 2 = 10$, $\text{distance}[j] = \text{distance}[E] = 11$; $10 < 11$

$\text{distance}[D] + \text{poids}(D,E) < \text{distance}[E]$

$\text{distance}[E] = \text{distance}[D] + \text{poids}(D,E) = 10$.

$\text{predecesseur}[E] \leftarrow D$;

Tableau des distances

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	4	3	7	8	10
2						
3						

Tableau des prédécesseurs

	A	B	C	D	E
init	Nil	Nil	Nil	Nil	Nil
1	S	C	A	C	D
2					
3					

La deuxième itération :

Pour tout arc (i, j) ∈ E faire

Si distance[i] + poids(i,j) < distance[j] alors

distance[j] ← distance[i] + poids(i,j);

predecesseur[j] ← i;

Arc (S,A) :

distance[S] + poids(S,A) = 0 +4= 4 , distance[j] = distance[A] = 4 ; 4 = 4

distance[S] + poids(S,A) = distance[A]

distance[A] = distance[S] + poids(S,A) =4, pas de changement

predecesseur[A] ← S;

Arc (S,B) :

distance[S] + poids(S,B) = 0 +6= 6 , distance[j] = distance[B] = 3 ; 6 > 3

distance[S] + poids(S,B) > distance[B]

distance[B] = 3, pas de changement

predecesseur[B] ← C;

Arc (A,C) :

distance[A] + poids(A,C) = 4 +3= 7 , distance[j] = distance[C] = 7 ; 7 =7

distance[A] + poids(A,C) = distance[C]

distance[C] = distance[A] + poids(A,C) =7, pas de changement

predecesseur[C] ← A;

Arc (B,A):

distance[B] + poids(B,A) = 3 +9= 12 , distance[j] = distance[A] = 4 ; 12 > 4

distance[A] =4 , pas de changement.

predecesseur[A] ← S; reste S.

Arc (B,D):

distance[B] + poids(B,D) = 3 + 4 = 7 , distance[j] = distance[D] = 8 ; 7 < 8

distance[B] + poids(B,D) < distance[D]

distance[D] = distance[B] + poids(B,D) =7

predecesseur[D] ← B;

Arc (C,B):

distance[C] + poids(C,B) = 7-4 = 3 , distance[j] = distance[B] = 3 ; 3 = 3

distance[C] + poids(C,B) = distance[B]

distance[B] = distance[C] + poids(C,B) =3, pas de changement.

predecesseur[B] ← C;

Arc (C,D):

distance[C] + poids(C,D) =7+1 =8 , distance[j] = distance[D] = 7 ; 8 >7

distance[C] + poids(C,D) > distance[D]

distance[D] = 7, pas de changement

predecesseur[D] ← B

Arc (C,E):

distance[C] + poids(C,E) = 7+4 = 11 , distance[j] = distance[E] = 10; 11 >10

distance[C] + poids(C,E) > distance[E]

distance[E] = 10, pas de changement

predecesseur[E] ← D;

Arc (D,E):

distance[D] + poids(D,E) = 7+2 = 9 , distance[j] = distance[E] = 10 ; 9 < 10

distance[D] + poids(D,E) < distance[E]

distance[E] = distance[D] + poids(D,E) = 9.

predecesseur[E] ← D;

Tableau des distances

	S	A	B	C	D	E
init	0	+∞	+∞	+∞	+∞	+∞
1	0	4	3	7	8	10
2	0	4	3	7	7	9
3						

Tableau des prédécesseurs

	A	B	C	D	E
init	Nil	Nil	Nil	Nil	Nil
1	S	C	A	C	D
2	S	C	A	B	D
3					

La troisième itération :

Pour tout arc $(i, j) \in E$ faire

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[j] \leftarrow \text{distance}[i] + \text{poids}(i,j)$;

$\text{predecesseur}[j] \leftarrow i$;

Arc (S,A) :

$\text{distance}[S] + \text{poids}(S,A) = 0 + 4 = 4$, $\text{distance}[j] = \text{distance}[A] = 4$; $4 = 4$

$\text{distance}[S] + \text{poids}(S,A) = \text{distance}[A]$

$\text{distance}[A] = \text{distance}[S] + \text{poids}(S,A) = 4$, pas de changement

$\text{predecesseur}[A] \leftarrow S$;

Arc (S,B) :

$\text{distance}[S] + \text{poids}(S,B) = 0 + 6 = 6$, $\text{distance}[j] = \text{distance}[B] = 3$; $6 > 3$

$\text{distance}[S] + \text{poids}(S,B) > \text{distance}[B]$

$\text{distance}[B] = 3$, pas de changement

$\text{predecesseur}[B] \leftarrow C$;

Arc (A,C) :

$\text{distance}[A] + \text{poids}(A,C) = 4 + 3 = 7$, $\text{distance}[j] = \text{distance}[C] = 7$; $7 = 7$

$\text{distance}[A] + \text{poids}(A,C) = \text{distance}[C]$

$\text{distance}[C] = \text{distance}[A] + \text{poids}(A,C) = 7$, pas de changement

$\text{predecesseur}[C] \leftarrow A$;

Arc (B,A):

$\text{distance}[B] + \text{poids}(B,A) = 3 + 9 = 12$, $\text{distance}[j] = \text{distance}[A] = 4$; $12 > 4$

$\text{distance}[A] = 4$, pas de changement.

$\text{predecesseur}[A] \leftarrow S$; reste S.

Arc (B,D):

$\text{distance}[B] + \text{poids}(B,D) = 3 + 4 = 7$, $\text{distance}[j] = \text{distance}[D] = 7$; $7 = 7$

$\text{distance}[B] + \text{poids}(B,D) = \text{distance}[D]$

$\text{distance}[D] = \text{distance}[B] + \text{poids}(B,D) = 7$, pas de changement

$\text{predecesseur}[D] \leftarrow B$;

Arc (C,B):

$\text{distance}[C] + \text{poids}(C,B) = 7 - 4 = 3$, $\text{distance}[j] = \text{distance}[B] = 3$; $3 = 3$

$\text{distance}[C] + \text{poids}(C,B) = \text{distance}[B]$

$\text{distance}[B] = \text{distance}[C] + \text{poids}(C,B) = 3$, pas de changement.

$\text{predecesseur}[B] \leftarrow C$;

Arc (C,D):

$\text{distance}[C] + \text{poids}(C,D) = 7 + 1 = 8$, $\text{distance}[j] = \text{distance}[D] = 7$; $8 > 7$

$\text{distance}[C] + \text{poids}(C,D) > \text{distance}[D]$

$\text{distance}[D] = 7$, pas de changement

$\text{predecesseur}[D] \leftarrow B$

Arc (C,E):

$\text{distance}[C] + \text{poids}(C,E) = 7 + 4 = 11$, $\text{distance}[j] = \text{distance}[E] = 9$; $11 > 9$

$\text{distance}[C] + \text{poids}(C,E) > \text{distance}[E]$

$\text{distance}[E] = 9$, pas de changement

$\text{predecesseur}[E] \leftarrow D$;

Arc (D,E):

$\text{distance}[D] + \text{poids}(D,E) = 7 + 2 = 9$, $\text{distance}[j] = \text{distance}[E] = 9$; $9 = 9$

$\text{distance}[D] + \text{poids}(D,E) = \text{distance}[E]$

$\text{distance}[E] = \text{distance}[D] + \text{poids}(D,E) = 9$, pas de changement.

$\text{predecesseur}[E] \leftarrow D$;

Tableau des distances

	S	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	4	3	7	8	10
2	0	4	3	7	7	9
3	0	4	3	7	7	9

Tableau des prédécesseurs

	A	B	C	D	E
init	Nil	Nil	Nil	Nil	Nil
1	S	C	A	C	D
2	S	C	A	B	D
3	S	C	A	B	D

On arrête ici car il n'y a pas de changement entre les étapes 2 et 3

Verification des circuits absorbants

Arc (u,v)	(S,A)	(S,B)	(A,C)	(B,A)	(B,D)	(C,B)	(C,D)	(C,E)	(D,E)
$d(u)+poids(u,v)$	4	6	7	12	7	3	8	11	9
$d(v)$	4	3	7	4	7	3	7	9	9

Pas de circuits absorbants

Remarque vous pouvez utiliser un seul tableau pour les distances et les prédecesseurs comme sur Dijkstra

Les plus courts chemins à partir de S sont :

- SA distance = 4
- SACB distance = 3
- SAC distance = 7
- SACBD distance = 7
- SACBDE distance = 9

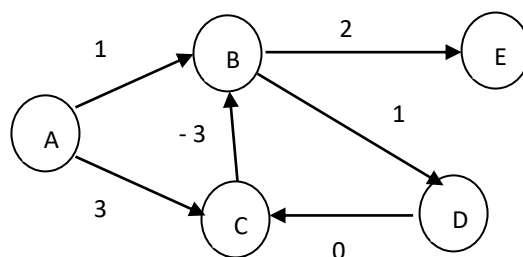
1. Compare the results of the two algorithms. What conclusion do you draw?

En comparant les résultats des 2 algorithmes on remarque que les distances sont différentes.

Par exemple la distance entre S et B est égale à 6 avec l'algorithme de Dijkstra et elle est égale à 3 avec l'algorithme de Bellman-Ford.

- L'algorithme de Dijkstra ne donne pas les plus courts chemins dans le cas de valeurs négatives
- L'algorithme de Bellman-Ford est le mieux adapté dans ce cas.

Exercise 2. Let the graph below:



1. What is the most appropriate algorithm for this graph? Justify your answer.

L'algorithme le plus approprié dans le cas de ce graphe est Bellman-Ford à cause des poids négatifs.

2. Show the execution of the chosen algorithm to find the shortest paths from point A.

Initialisation :

Ordre d'examen des arcs :

Arc	(A,B)	(A,C)	(B,D)	(B,E)	(C,B)	(D,C)
Poids	1	3	1	2	-3	0

Tableau des distances

	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1					
2					
3					
4					

Tableau des prédécesseurs

	B	C	D	E
init	Nil	Nil	Nil	Nil
1				
2				
3				
4				

La première itération :

Pour tout arc $(i, j) \in E$ faire

Si $\text{distance}[i] + \text{poids}(i,j) < \text{distance}[j]$ alors

$\text{distance}[j] \leftarrow \text{distance}[i] + \text{poids}(i,j);$

$\text{predecesseur}[j] \leftarrow i;$

Arc (A,B) :

$\text{distance}[A] + \text{poids}(A,B) = 0 + 1 = 1$, $\text{distance}[j] = \text{distance}[B] = +\infty$; $1 < +\infty$

$\text{distance}[A] + \text{poids}(A,B) < \text{distance}[B]$

$\text{distance}[B] = \text{distance}[A] + \text{poids}(A,B) = 1$

$\text{predecesseur}[B] \leftarrow A;$

Arc (A,C) :

$\text{distance}[A] + \text{poids}(A,C) = 0 + 3 = 3$, $\text{distance}[j] = \text{distance}[C] = +\infty$; $3 < +\infty$

$\text{distance}[A] + \text{poids}(A,C) < \text{distance}[C]$

$\text{distance}[C] = \text{distance}[A] + \text{poids}(A,C) = 3$

$\text{predecesseur}[C] \leftarrow A;$

Arc (B,D) :

$\text{distance}[B] + \text{poids}(B,D) = 1 + 1 = 2$, $\text{distance}[j] = \text{distance}[D] = +\infty$; $2 < +\infty$

$\text{distance}[B] + \text{poids}(B,D) < \text{distance}[D]$

$\text{distance}[D] = \text{distance}[B] + \text{poids}(B,D) = 2$

$\text{predecesseur}[D] \leftarrow B;$

Arc (B,E) :

$\text{distance}[B] + \text{poids}(B,E) = 1 + 2 = 3$, $\text{distance}[j] = \text{distance}[E] = +\infty$; $3 < +\infty$

$\text{distance}[B] + \text{poids}(B,E) < \text{distance}[E]$

$\text{distance}[E] = \text{distance}[B] + \text{poids}(B,E) = 3$

$\text{predecesseur}[E] \leftarrow B;$

Arc (C,B) :

$\text{distance}[C] + \text{poids}(C,B) = 3 - 3 = 0$, $\text{distance}[j] = \text{distance}[B] = 1$; $0 < 1$

$\text{distance}[C] + \text{poids}(C,B) < \text{distance}[B]$

$\text{distance}[B] = \text{distance}[C] + \text{poids}(C,B) = 0$

$\text{predecesseur}[B] \leftarrow C;$

Arc (D,C) :

$\text{distance}[D] + \text{poids}(D,C) = 2 + 0 = 2$, $\text{distance}[j] = \text{distance}[C] = 3$; $2 < 3$

$\text{distance}[D] + \text{poids}(D,C) < \text{distance}[C]$

$\text{distance}[C] = \text{distance}[D] + \text{poids}(D,C) = 2$

$\text{predecesseur}[C] \leftarrow D;$

Tableau des distances

	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	0	2	2	3
2					
3					
4					

Tableau des prédécesseurs

	B	C	D	E
init	Nil	Nil	Nil	Nil
1	C	D	B	B
2				
3				
4				

La deuxième itération :

Pour tout arc (i, j) ∈ E faire

Si distance[i] + poids(i,j) < distance[j] alors

distance[j] ← distance[i] + poids(i,j);

predecesseur[j] ← i;

Arc (A,B) :

distance[A] + poids(A,B) = 0 +1= 1 , distance[j] = distance[B] = 0 ; 1 > 0

distance[A] + poids(A,B) > distance[B]

distance[B] = 0, pas de changement.

predecesseur[B] ← C;

Arc (A,C) :

distance[A] + poids(A,C) = 0 +3= 3 , distance[j] = distance[C] = 2 ; 3 > 2

distance[A] + poids(A,C) > distance[C]

distance[C] = 2, pas de changement.

predecesseur[C] ← D;

Arc (B,D) :

distance[B] + poids(B,D) = 0 + 1= 1 , distance[j] = distance[D] = 2 ; 1 < 2

distance[B] + poids(B,D) < distance[D]

distance[D] = distance[B] + poids(B,D) = 1

predecesseur[D] ← B;

Arc (B,E) :

distance[B] + poids(B,E) = 0 +2= 2 , distance[j] = distance[E] = 3 ; 2 < 3

distance[B] + poids(B,E) < distance[E]

distance[E] = distance[B] + poids(B,E) = 2

predecesseur[E] ← B;

Arc (C,B) :

distance[C] + poids(C,B) = 2 -3= -1 , distance[j] = distance[B] = 0; -1 < 0

distance[C] + poids(C,B) < distance[B]

distance[B] = distance[A] + poids(A,C) = - 1

predecesseur[B] ← C;

Arc (D,C) :

distance[D] + poids(D,C) = 1 +0= 1 , distance[j] = distance[C] = 2 ; 1 < 2

distance[D] + poids(D,C) < distance[C]

distance[C] = distance[D] + poids(D,C) = 1

predecesseur[C] ← D;

Tableau des distances

	A	B	C	D	E
init	0	+∞	+∞	+∞	+∞
1	0	0	2	2	3
2	0	-1	1	1	2
3					
4					

Tableau des prédécesseurs

	B	C	D	E
init	Nil	Nil	Nil	Nil
1	C	D	B	B
2	C	D	B	B
3				
4				

La troisième itération :

Pour tout arc (i, j) ∈ E faire

Si distance[i] + poids(i,j) < distance[j] alors

distance[j] ← distance[i] + poids(i,j);

predecesseur[j] ← i;

Arc (A,B) :

distance[A] + poids(A,B) = 0 +1= 1 , distance[j] = distance[B] = -1 ; 1 > -1

distance[A] + poids(A,B) > distance[B]

distance[B] = - 1, pas de changement.

predecesseur[B] ← C;

Arc (A,C) :

distance[A] + poids(A,C) = 0 +3= 3 , distance[j] = distance[C] = 1 ; 3 > 1

distance[A] + poids(A,C) > distance[C]

distance[C] = 1, pas de changement.

predecesseur[C] ← D;

Arc (B,D) :

distance[B] + poids(B,D) = -1 + 1 = 0 , distance[j] = distance[D] = 1 ; 0 < 1

distance[B] + poids(B,D) < distance[D]

distance[D] = distance[B] + poids(B,D) = 0

predecesseur[D] ← B;

Arc (B,E) :

distance[B] + poids(B,E) = -1 + 2 = 1 , distance[j] = distance[E] = 2 ; 1 < 2

distance[B] + poids(B,E) < distance[E]

distance[E] = distance[B] + poids(B,E) = 1

predecesseur[E] ← B;

Arc (C,B) :

distance[C] + poids(C,B) = 1 - 3 = -2 , distance[j] = distance[B] = -1; -2 < -1

distance[C] + poids(C,B) < distance[B]

distance[B] = distance[A] + poids(A,C) = -2

predecesseur[B] ← C;

Arc (D,C) :

distance[D] + poids(D,C) = 0 + 0 = 0 , distance[j] = distance[C] = 1 ; 0 < 1

distance[D] + poids(D,C) < distance[C]

distance[C] = distance[D] + poids(D,C) = 0

predecesseur[C] ← D;

Tableau des distances

	A	B	C	D	E
init	0	+∞	+∞	+∞	+∞
1	0	0	2	2	3
2	0	-1	1	1	2
3	0	-2	0	0	1
4					

Tableau des prédécesseurs

	B	C	D	E
init	Nil	Nil	Nil	Nil
1	C	D	B	B
2	C	D	B	B
3	C	D	B	B
4				

La quatrième itération :

Pour tout arc (i, j) ∈ E faire

Si distance[i] + poids(i,j) < distance[j] alors

distance[j] ← distance[i] + poids(i,j);

predecesseur[j] ← i;

Arc (A,B) :

distance[A] + poids(A,B) = 0 + 1 = 1 , distance[j] = distance[B] = -2 ; 1 > -2

distance[A] + poids(A,B) > distance[B]

distance[B] = -2, pas de changement.

predecesseur[B] ← C;

Arc (A,C) :

distance[A] + poids(A,C) = 0 + 3 = 3 , distance[j] = distance[C] = 0 ; 3 > 0

distance[A] + poids(A,C) > distance[C]

distance[C] = 0, pas de changement.

predecesseur[C] ← D;

Arc (B,D) :

distance[B] + poids(B,D) = -2 + 1 = -1 , distance[j] = distance[D] = 0 ; -1 < 0

distance[B] + poids(B,D) < distance[D]

distance[D] = distance[B] + poids(B,D) = -1

predecesseur[D] ← B;

Arc (B,E) :

distance[B] + poids(B,E) = -2 + 2 = 0 , distance[j] = distance[E] = 1 ; 0 < 1

distance[B] + poids(B,E) < distance[E]

distance[E] = distance[B] + poids(B,E) = 0

predecesseur[E] ← B;

Arc (C,B) :

$\text{distance}[C] + \text{poids}(C,B) = 0 - 3 = -3$, $\text{distance}[j] = \text{distance}[B] = -2$; $-3 < -2$

$\text{distance}[C] + \text{poids}(C,B) < \text{distance}[B]$

$\text{distance}[B] = \text{distance}[A] + \text{poids}(A,C) = -3$

$\text{predecesseur}[B] \leftarrow C$;

Arc (D,C) :

$\text{distance}[D] + \text{poids}(D,C) = -1 + 0 = -1$, $\text{distance}[j] = \text{distance}[C] = 0$; $-1 < 0$

$\text{distance}[D] + \text{poids}(D,C) < \text{distance}[C]$

$\text{distance}[C] = \text{distance}[D] + \text{poids}(D,C) = -1$

$\text{predecesseur}[C] \leftarrow D$;

Tableau des distances

	A	B	C	D	E
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	0	2	2	3
2	0	-1	1	1	2
3	0	-2	0	0	1
4	0	-3	-1	-1	0

Tableau des prédécesseurs

	B	C	D	E
init	Nil	Nil	Nil	Nil
1	C	D	B	B
2	C	D	B	B
3	C	D	B	B
4	C	D	B	B

(A,B)	(A,C)	(B,D)	(B,E)	(C,B)	(D,C)
1	3	1	2	-3	0

Verification des circuits absorbants

Arc (u,v)	(A,B)	(A,C)	(B,D)	(B,E)	(C,B)	(D,C)
$d(u) + \text{poids}(u,v)$	1	3	-1	0	-3	-1
$d(v)$	-3	-1	-1	0	-3	-1

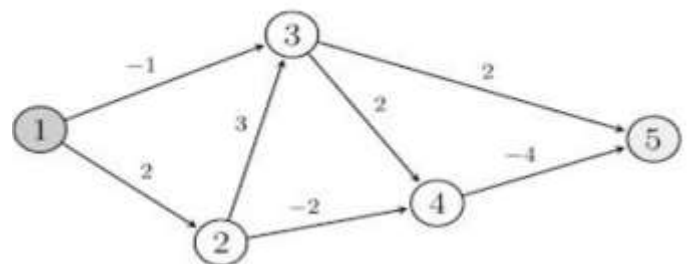
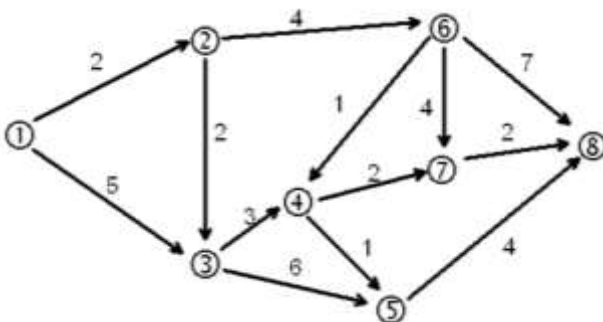
Le changement ne s'arrête pas il existe un circuit absorbant

Après exécution on trouve un circuit absorbant

3. What are the shortest paths from point A?

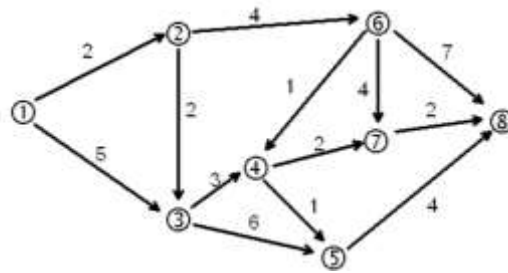
Il n'y a pas de plus courts chemins à partir de A

Exercise 3. Apply Dijkstra's and Bellman-Ford's algorithms to find the shortest paths for each of the following graphs:



1-

2- Exercice 3 :



Le graphe contient des valeurs positives alors on peut utiliser l'algorithme de Dijkstra

	1	2	3	4	5	6	7	8
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1		2 - 1	5 - 1	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
2			4 - 2	$+\infty$	$+\infty$	6 - 2	$+\infty$	$+\infty$
3				7 - 3	10 - 3	6 - 2	$+\infty$	$+\infty$
4				7 - 3	10 - 3		10 - 6	13 - 6
5					8 - 4		9 - 4	13 - 6
6							9 - 4	12 - 4
7								11 - 7

Algorithme de Bellman-Ford

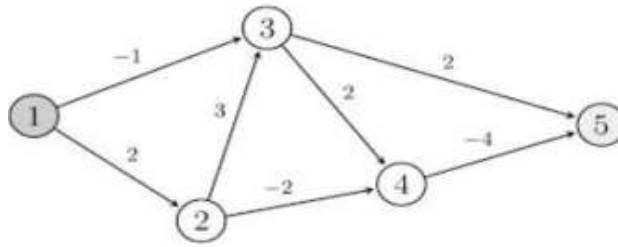
	1	2	3	4	5	6	7	8
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	2	4	7	8	6	9	11
2	0	2	4	7	8	6	9	11

	2	3	4	5	6	7	8
init	Nil	Nil	Nil	Nil	Nil	Nil	Nil
1	1	2	3	4	2	4	7
2	1	2	3	4	2	4	7

Pas de circuits absorbants

Remarque : les 2 algorithmes donnent les mêmes résultats.

- Le graphe suivant contient des poids négatifs donc on utilisera l'algorithme de Bellman-Ford.



Ordre d'examen des arcs :

Arc	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	(3,5)	(4,5)
Poids	2	-1	3	-2	2	2	-4

Tableau des distances

	1	2	3	4	5
init	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$
1	0	2	-1	0	-4
2	0	2	-1	0	-4

Arrêt il n'y a pas de changement entre les étapes 1 et 2

Tableau des prédécesseurs

	2	3	4	5
init	Nil	Nil	Nil	Nil
1	1	1	2	4
2	1	1	2	4

Verification des circuits absorbants

Arc (u,v)	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	(3,5)	(4,5)
$d(u)+poids(u,v)$	2	-1	5	0	1	1	-4
$d(v)$	2	-1	-1	0	0	-4	-4

Pas de circuits absorbants

Remarques.

- L'algorithme de Dijkstra permet de trouver l'arbre des plus courts chemins à partir d'un sommet source.
- L'algorithme de Dijkstra ne marche qu'avec les graphes dont les poids sont positifs.
- L'algorithme de Bellman-Ford retourne bien une arborescence de plus court chemin entre s (source) et les autres sommets dans un graphe sans circuit et dont les poids peuvent être négatifs.
- Si le graphe comporte n sommets, et s'il ne contient pas de circuit absorbant, un plus court chemin sera de longueur inférieure à n et au bout de $n-1$ passages, on aura trouvé tous les plus courts chemins partant de s_0 .
- Arrêter l'algorithme de Bellman-Ford et retourner les chemins les plus courts si les distances ne sont pas modifiées d'une itération à l'autre.