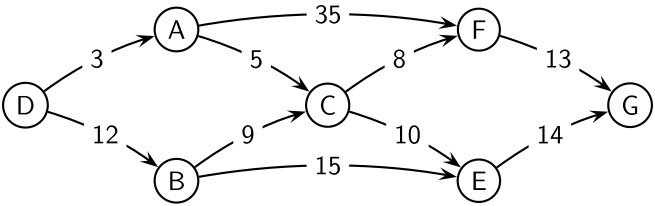
Algorithme de Dijkstra

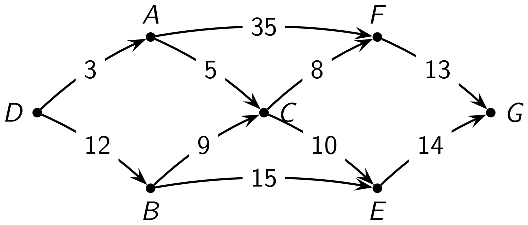
Algorithme de recherche d'une plus courte chaîne d'un graphe pondéré.
On cherche à déterminer le plus court chemin entre le sommet de départ D et le sommet d'arrivée G.



Algorithme de Dijkstra sur le graphe

Etape initiale

On marque :

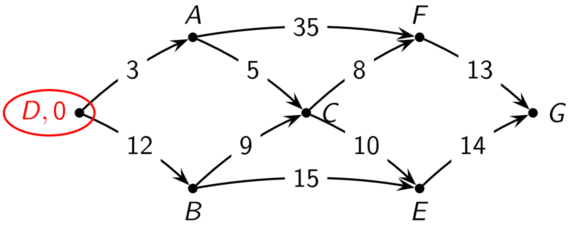


Algorithme de Dijkstra sur le graphe

Etape initiale

On marque :

- le sommet de départ avec un poids nul : $D, 0$

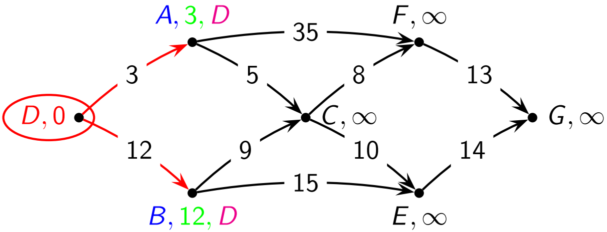


Algorithme de Dijkstra sur le graphe

Etape initiale

On marque :

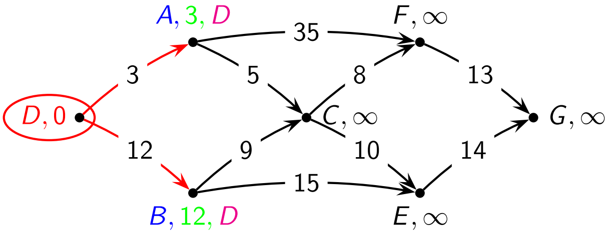
- les sommets adjacents avec leurs poids et le sommet de provenance : $A, 3, D$ et $B, 12, D$.



Algorithme de Dijkstra sur le graphe

Etape initiale

On marque :

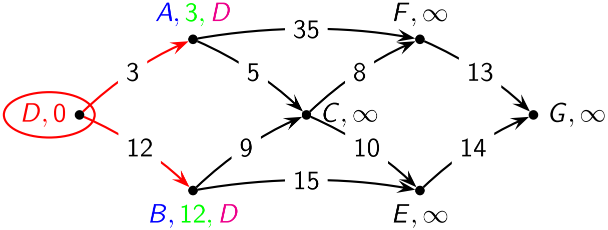


On dit que le sommet D est fixé . Quand tous les sommets sont fixés l'algorithme s'arrête.

Algorithme de Dijkstra sur le graphe

Première itération

On applique les opérations suivantes :

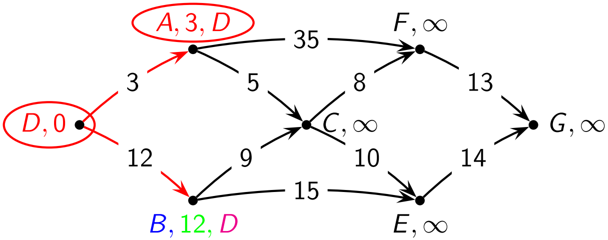


Algorithme de Dijkstra sur le graphe

Première itération

On applique les opérations suivantes :

- On fixe le sommet de plus petit poids en précisant le sommet de provenance : $A, 3, D$

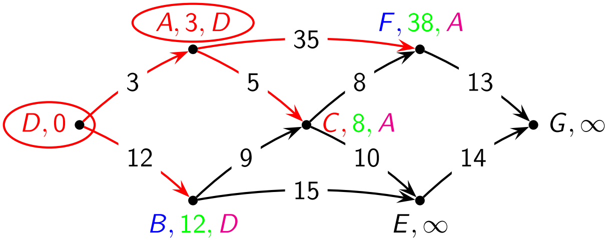


Algorithme de Dijkstra sur le graphe

Première itération

On applique les opérations suivantes :

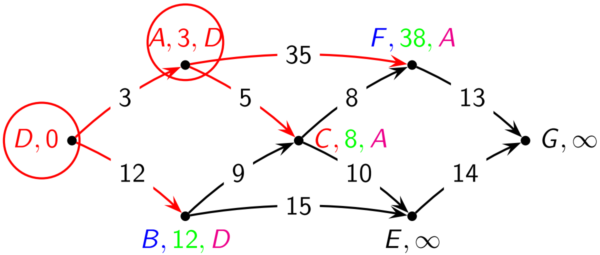
- On marque les sommets adjacents **non fixés** avec la somme des poids parcourus et le sommet de provenance : $F, 38, A$ et $C, 8, A$.



Algorithme de Dijkstra sur le graphe

Deuxième itération

On réitère les mêmes opérations :

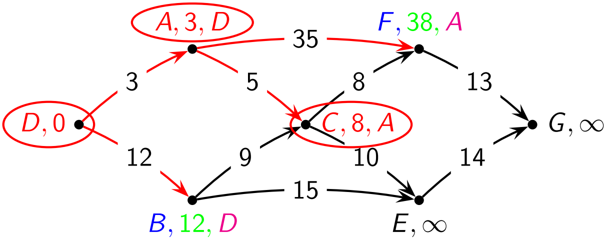


Algorithme de Dijkstra sur le graphe

Deuxième itération

On réitère les mêmes opérations :

- On fixe le sommet de plus petit poids en précisant le sommet de provenance : $C, 8, D$

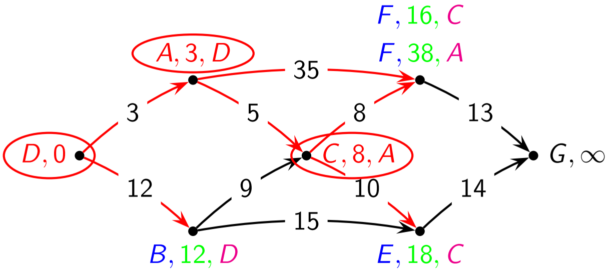


Algorithme de Dijkstra sur le graphe

Deuxième itération

On réitère les mêmes opérations :

- On marque les sommets adjacents non fixés avec la somme des poids parcourus et le sommet de provenance : $F, 16, C$ et $E, 18, C$.

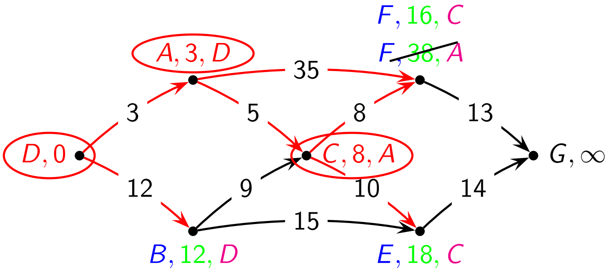


Algorithme de Dijkstra sur le graphe

Deuxième itération

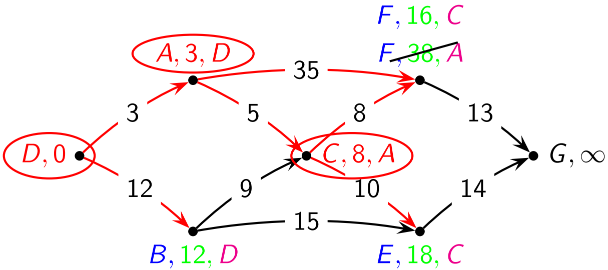
On réitère les mêmes opérations :

- Si une marque précédente a un poids supérieur à une nouvelle marque obtenue, on barre l'ancienne marque.



Algorithme de Dijkstra sur le graphe

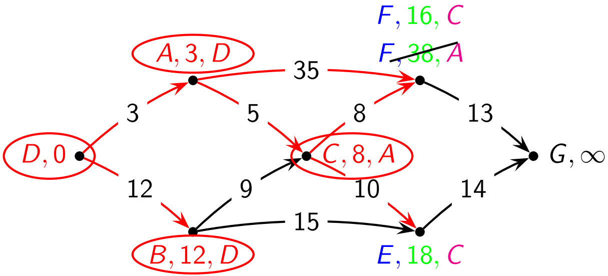
Troisième itération



Algorithme de Dijkstra sur le graphe

Troisième itération

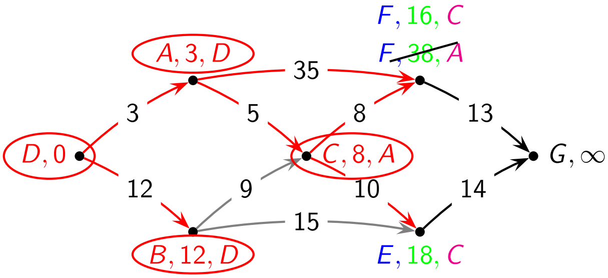
- le sommet de plus petit poids est : $B, 12, D$



Algorithme de Dijkstra sur le graphe

Troisième itération

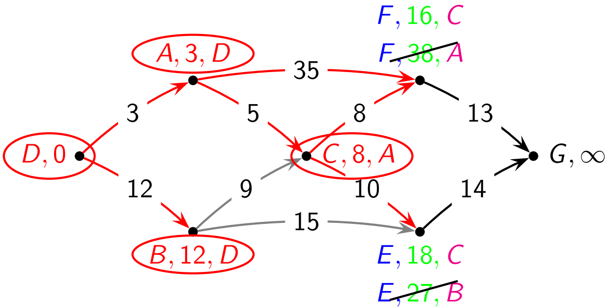
- Le seul sommet adjacent non fixé est : $E, 27, B$.



Algorithme de Dijkstra sur le graphe

Troisième itération

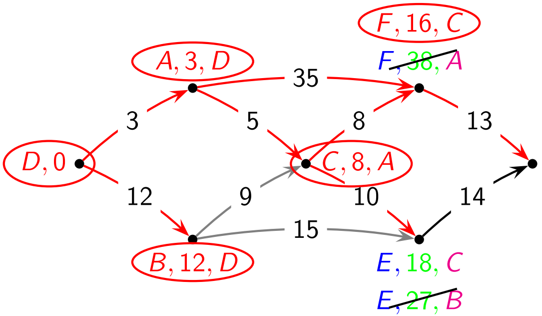
- La nouvelle marque obtenue a un poids supérieur à la marque précédente $E, 18, C$. On laisse donc la précédente marque.



Algorithme de Dijkstra sur le graphe

Quatrième itération

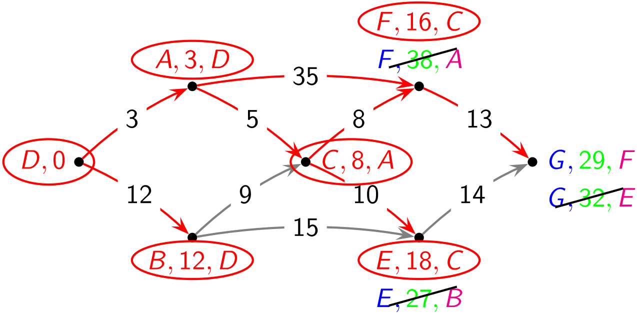
- le sommet de plus petit poids est : $F, 16, C$
- Il y a un seul sommet adjacent non marqué (donc non fixé) : $G, 29, F$.



Algorithme de Dijkstra sur le graphe

Cinquième itération

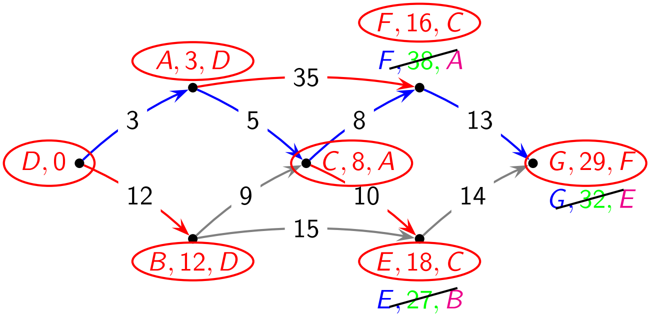
- le sommet de plus petit poids est : $E, 18, C$
- Il y a un seul sommet adjacent non fixé : $G, 32, E$.
- La nouvelle marque obtenue a un poids supérieur à la marque précédente $G, 29, F$. On laisse donc la précédente marque.



Algorithme de Dijkstra sur le graphe

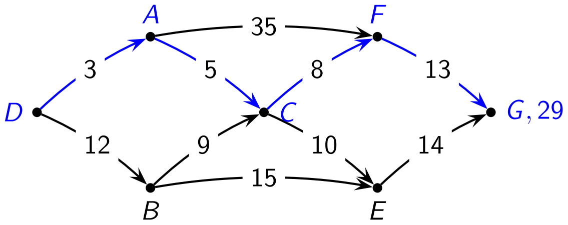
Sixième itération

On fixe G avec le poids 29. $S = \{D, A, C, B, F, E, G\}$. Tous les sommets étant fixés, on peut reconstituer le chemin le plus court en remontant la chaîne ainsi obtenue.



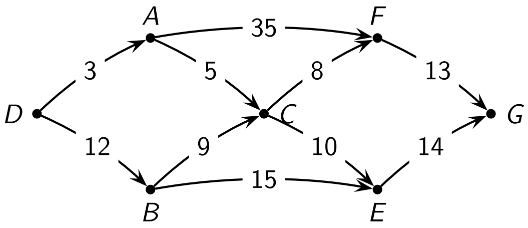
Algorithme de Dijkstra sur le graphe

Le chemin le plus court est donc : $D - A - C - F - G$**, de poids 29.**



Algorithme de Dijkstra dans un tableau

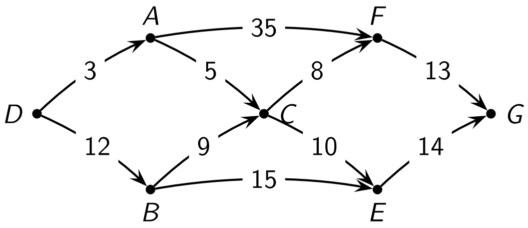
Etape initiale



On peut aussi décrire l'application de l'algorithme de Dijkstra en réalisant un tableau.

Algorithme de Dijkstra dans un tableau

Etape initiale

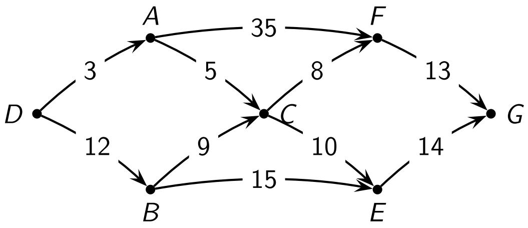


Sur la première ligne, on écrit les sommets du graphe, en commençant par le sommet du départ D, les sommets adjacents à D, les autres sommets en terminant par le sommet de fin.

(D)	A	B	C	E	F	G

Algorithme de Dijkstra dans un tableau

Etape initiale

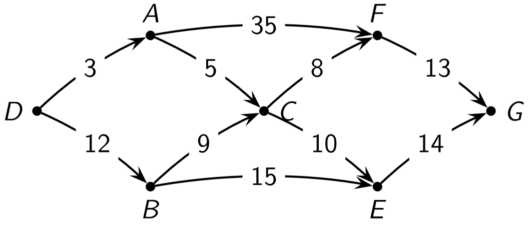


Sur la deuxième ligne, on écrit en rouge, le poids nul pour le sommet de départ et on entoure le sommet (D) pour indiquer qu'il est fixé.

(D)	A	B	C	E	F	G
0						

Algorithme de Dijkstra dans un tableau

Etape initiale

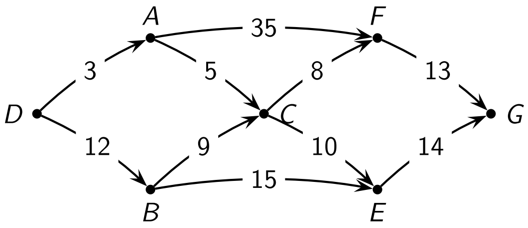


Le poids et le sommet de provenance pour les sommets adjacents.

(D)	A	B	C	E	F	G
0	0+3 = 3 (D)	0+12 = 12 (D)				

Algorithme de Dijkstra dans un tableau

Etape initiale

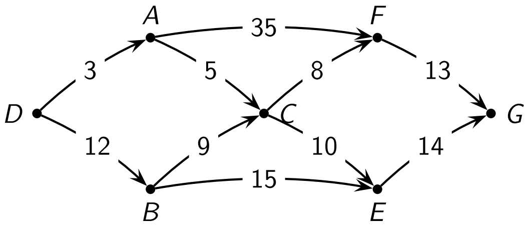


Le symbole infini pour les autres sommets.

(D)	A	B	C	E	F	G
0	0+3 = 3 (D)	0+12 = 12 (D)	∞	∞	∞	∞

Algorithme de Dijkstra dans un tableau

Première itération

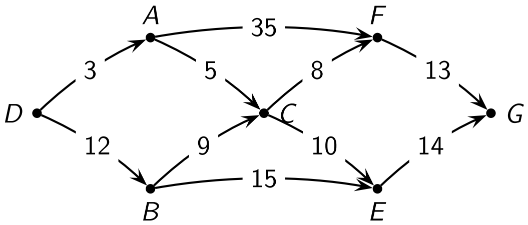


Après avoir chosit le sommet non fixé de plus faible poids, A, on l'entoure.

(D)	(A)	B	C	E	F	G

Algorithme de Dijkstra dans un tableau

Première itération

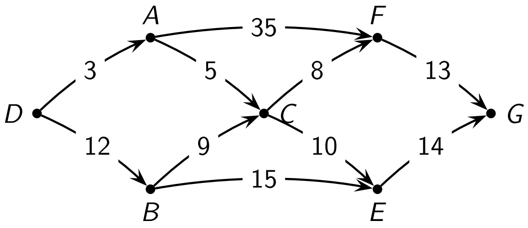


On note en rouge dans sa colonne le poids de la chaîne y conduisant ainsi que le sommet de provenance.

(D)	(A)	B	C	E	F	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	$3(D)$					

Algorithme de Dijkstra dans un tableau

Première itération

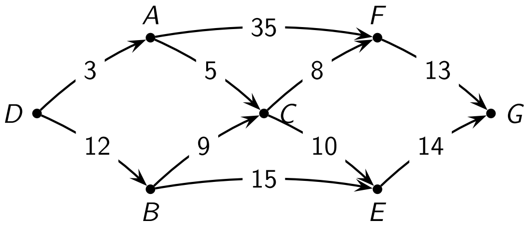


Dans la colonne des sommets adjacents à A, on ajoute les poids chaînes y conduisant avec le rappel du sommet fixé D d'origine.

(D)	(A)	B	C	E	F	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	$3(D)$		$3+5=8$ (A)		$3+35=$ 38(A)	

Algorithme de Dijkstra dans un tableau

Première itération

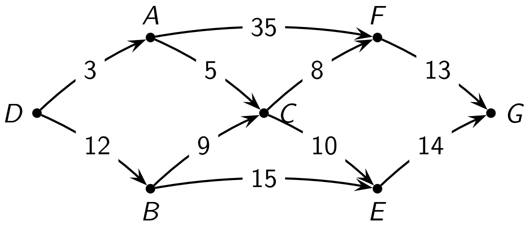


On complète en réécrivant les indications de la ligne précédente.

(D)	(A)	B	C	E	F	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	$3(D)$	12 (D)	$3+5=8$ (A)	∞	$3+35=$ 38(A)	∞

Algorithme de Dijkstra dans un tableau

deuxième itération

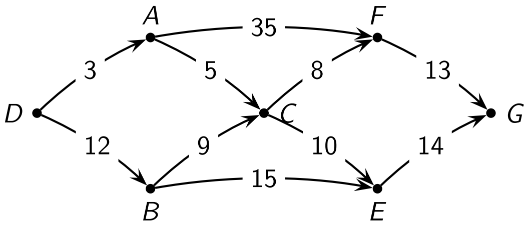


Le nouveau sommet fixé est C. Ses sommets adjacants sont alors marqués.

(D)	(A)	B	(C)	E	F	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	3(D)	12 (D)	8 (A)	∞	38(A)	∞

Algorithme de Dijkstra dans un tableau

deuxième itération

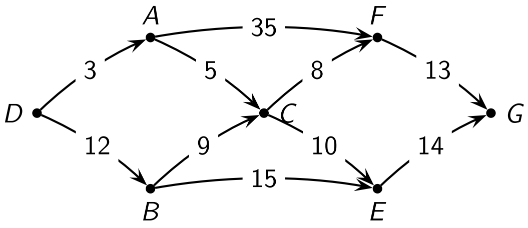


Le nouveau sommet fixé est C. Ses sommets adjacants sont alors marqués.

(D)	(A)	B	(C)	E	F	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	3(D)	12 (D)	8 (A)	∞	38(A)	∞
			8(A)			

Algorithme de Dijkstra dans un tableau

deuxième itération

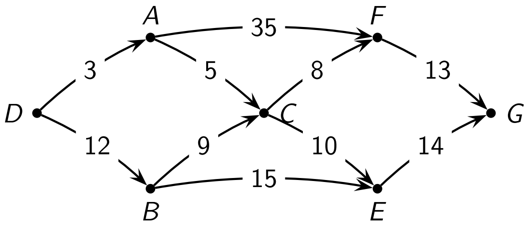


Le nouveau sommet fixé est C. Ses sommets adjacants sont alors marqués.

(D)	(A)	B	(C)	E	F	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	3(D)	12 (D)	8 (A)	∞	38(A)	∞
			8(A)	$8+10=18$ (C)	$8+8=16$ (C)	

Algorithme de Dijkstra dans un tableau

deuxième itération

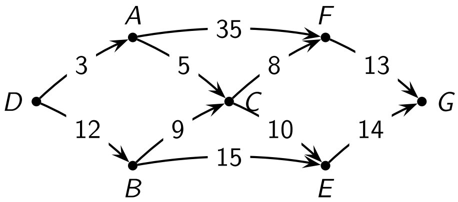


Le nouveau sommet fixé est C. Ses sommets adjacants sont alors marqués.

(D)	(A)	B	(C)	E	F	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	3(D)	12 (D)	8 (A)	∞	38(A)	∞
		12 (D)	8(A)	$8+10=18$ (C)	$8+8=16$ (C)	∞

Algorithme de Dijkstra dans un tableau

troisième itération

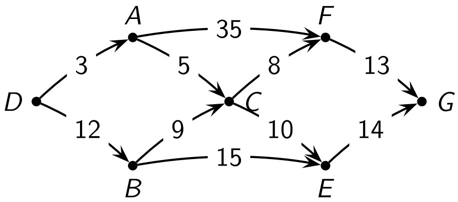


Le nouveau sommet fixé est B.

(D)	(A)	(B)	(C)	E	F	G
0	$0+3=3$ (D)	$0+12=12$ (D)	∞	∞	∞	∞
	$3(D)$	12 (D)	8 (A)	∞	38(A)	∞
		12 (D)	$8(A)$	18 (C)	16 (C)	∞
		$12(D)$				

Algorithme de Dijkstra dans un tableau

troisième itération

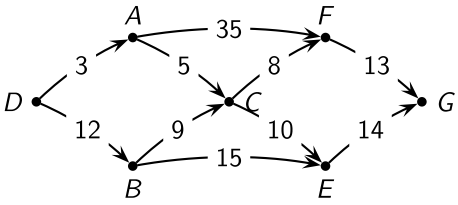


Le sommet adjacent C est déjà fixé et donc exclue des calculs.

(D)	(A)	(B)	(C)	E	F	G
0	$0+3=3$ (D)	$0+12=12$ (D)	∞	∞	∞	∞
	$3(D)$	12 (D)	8 (A)	∞	38(A)	∞
		12 (D)	$8(A)$	18 (C)	16 (C)	∞
		$12(D)$				∞

Algorithme de Dijkstra dans un tableau

troisième itération

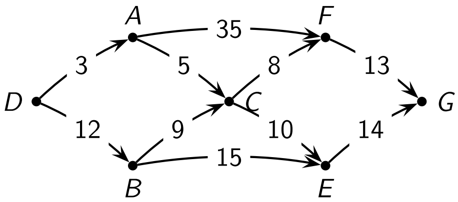


Pour le sommet adjacent E, le poids de la chaîne passant par B est supérieur à celui de celle vanant de C. On conserve donc cette dernière.

(D)	(A)	(B)	(C)	E	F	G
0	$0+3=3$ (D)	$0+12=12$ (D)	∞	∞	∞	∞
	3(*D*)	12 (D)	8 (A)	∞	38(A)	∞
		12 (D)	8(*A*)	18 (C)	16 (C)	∞
		12(*D*)		$12+15=$ $27>18$ 18(C)		∞

Algorithme de Dijkstra dans un tableau

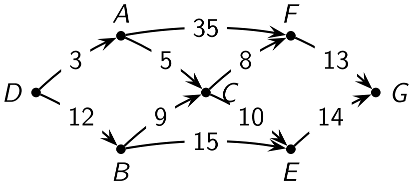
troisième itération



(D)	(A)	(B)	(C)	E	F	G
0	$0+3=3$ (D)	$0+12=12$ (D)	∞	∞	∞	∞
	$3(D)$	12 (D)	8 (A)	∞	38(A)	∞
		12 (D)	$8(A)$	18 (C)	16 (C)	∞
		$12(D)$		$12+15=27>18$ 18(C)	16 (C)	∞

Algorithme de Dijkstra dans un tableau

quatrième itération

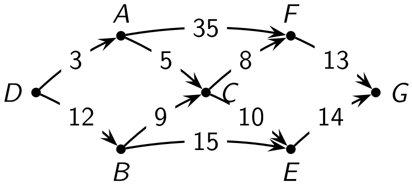


Le nouveau sommet fixé est F.

(D)	(A)	(B)	(C)	E	(F)	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	3(D)	12 (D)	8 (A)	∞	38 (A)	∞
		12(D)	8(A)	18 (C)	16 (C)	∞
		12(D)		18 (C)	16 (C)	∞
					16(C)	

Algorithme de Dijkstra dans un tableau

quatrième itération

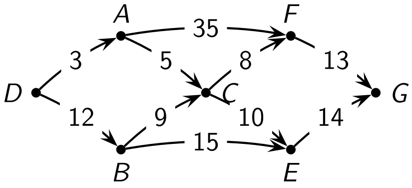


G est son unique sommet adjacent.

(D)	(A)	(B)	(C)	E	(F)	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	3(*D*)	12 (D)	8 (A)	∞	38 (A)	∞
		12(D)	8(*A*)	18 (C)	16 (C)	∞
		12(*D*)		18 (C)	16 (C)	∞
					16(*C*)	$16 + 13$ $= 19$(F)

Algorithme de Dijkstra dans un tableau

quatrième itération



(D)	(A)	(B)	(C)	E	(F)	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	3(*D*)	12 (D)	8 (A)	∞	38 (A)	∞
		12(D)	8(*A*)	18 (C)	16 (C)	∞
		12(*D*)		18 (C)	16 (C)	∞
				18 (C)	16(*C*)	$16 + 13$ $= 19$(F)

Algorithme de Dijkstra dans un tableau

cinquième itération

Pour son seul sommet adjacent G, le poids de la chaîne passant par E est supérieur à celui de celle vanant de F. On conserve donc cette dernière.

(D)	(A)	(B)	(C)	(E)	(F)	G
0	3 (D)	12 (D)	∞	∞	∞	∞
	$3(D)$	12 (D)	8 (A)	∞	38 (A)	∞
		12(D)	$8(A)$	18 (C)	16 (C)	∞
		$12(D)$		18 (C)	16 (C)	∞
				18 (C)	16 (C)	29 (F)
				$18(C)$		$18+14=$ $32>29$ 29(F)

Algorithme de Dijkstra dans un tableau

dernière itération et conclusion

(D)	(A)	(B)	(C)	(E)	(F)	(G)
0	3 (D)	12 (D)	∞	∞	∞	∞
	$3(D)$	12 (D)	8 (A)	∞	38 (A)	∞
		12(D)	$8(A)$	18 (C)	16 (C)	∞
		$12(D)$		18 (C)	16 (C)	∞
				18 (C)	16 (C)	29 (F)
				$18(C)$		29(F)
						$29(F)$

Conclusion : Le trajet le plus court est D-A-C-F-G