



TP N°4: **PROBLEME DE** **PCC ET** **COLORATION**

 [Visual graph tree](#)

PRÉPARER PAR
IDRISS ABDELMAGHNI KEBLADJ.
BOUCHACHI MEHDI MOHAMED.

RESPONSABLE DU TP:
MME S.AROUSSI

ENVIRONNEMENT UTILISÉ

👉 **Processeur (CPU)**



I7 5eme Gen

👉 **Mémoire vive (RAM)**

16 GB DDR4

👉 **système d'exploitation**



👉 **Langage de programmation**

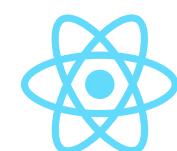


👉 **Environnement**



Visual Studio Code

👉 **Bibliothèques**



React

👉 **Other tools**

React icons/ React hot toast/ Styled-components/ Assistant Utilisée, ChatGPT 5%

Assure la rapidité et la stabilité lors du développement et de la compilation.

Permet d'exécuter plusieurs outils et serveurs sans ralentissement.

Fournit un environnement fiable et compatible pour le développement web.

Utilisé pour créer des interfaces interactives et dynamiques.

Outil principal d'édition et de gestion du code source.

Ensemble de modules facilitant la création d'applications réactives.

👉 **Plus court chaîne(chemin) :**

- ❖ Introduction et Définition.
- ❖ Algorithme.
- ❖ Principe de fonctionnement.
- ❖ Implémentation.
- ❖ complexité.

👉 **problème de coloration des sommets:**

- ❖ Introduction et Définition.
- ❖ Algorithme.
- ❖ Principe de fonctionnement.
- ❖ Implémentation.
- ❖ complexité.

Plus court chaîne(chemin)

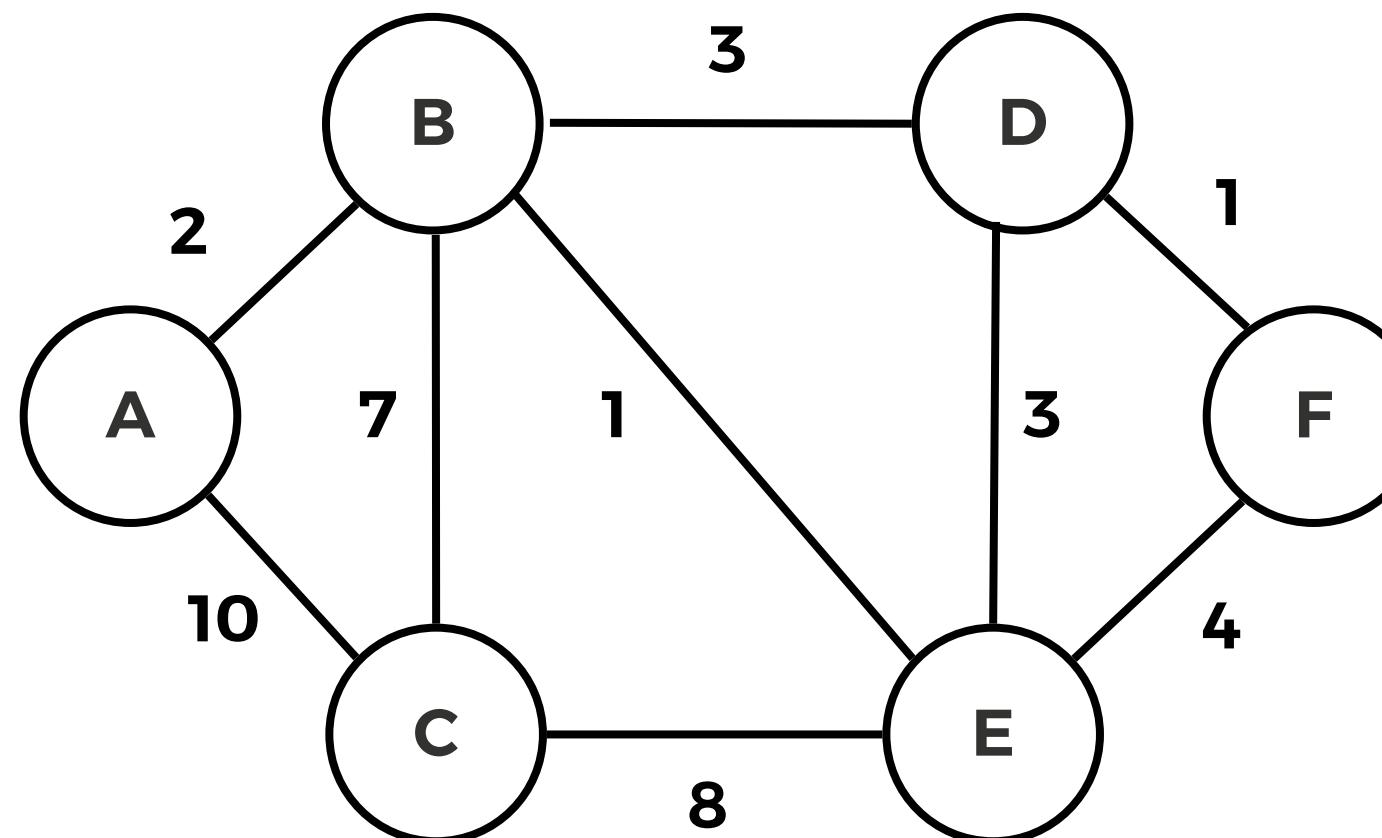
✿ INTRODUCTION.

- 👉 Nous aimerais disposer d'une méthode capable, pour tout graphe et pour toute paire de sommets s et t , de d'exterminer le plus court chemin (chaîne)entre eux. Résoudre ce problème va donc consister `a proposer un algorithme, aussi rapide que possible.
- 👉 nous allons généraliser dans le cas de graphe `values` , où chaque arête(arce) est associée à une valeur, appelé souvent son poids, $c(e)$: le chemin(chaîne) de poids minimum, celui dont la somme des poids des arêtes est le plus faible possible.
L'algorithme de Dijkstra est l'un de ceux qui convient le mieux

❖ DÉFINITION.

👉 Plus court chaîne(chemin) dans un graphe valué:

- Dans un graphe value, le poids $c(p)$ d'une chaîne p est la somme des poids des arêtes le long du chemin
- Dans ce qui suit nous appellerons le poids d'une chaîne sa longueur. Le plus court chaîne entre 2 sommets A et F est alors défini comme la chaîne de plus faible poids reliant A et F .



❖ ALGORITHME.

pour trouver les plus court chemins(chaînes) dans graphe valué (orienté ou non),nous allons voir l'algorithme suivant :

👉 **Algorithme de Dijkstra:**

Cet algorithme est une adaptation de l'algorithme de recherche pour calculer les plus courts chemins(chaînes) d'un sommet s à tous les autres sommets du graphe.

✿ PRINCIPE DE FONCTIONNEMENT.

💡 Graphe orienté:

- 👉 Soit $\pi(i)$ la valeur de chemin du sommet « s » vers le sommet « i », ainsi, initialement : $\pi(s) = 0$ et $\pi(x) = \infty$ pour tout sommet $x \neq s$. Soit M l'ensemble des sommets marqués, initialement il est vide ($M = \varnothing$)
- 👉 Tant qu'il existe un sommet non marqué ($M \neq X$) ou on n'est pas arrivé au sommet destinataire ($x \neq d$) faire:
 1. Choisir un sommet non marqué, soit x ($x \in X - M$), ayant le plus petit π [i.e. $\pi(x) = \min \{\pi(y) \text{ tq } y \in X - M\}$]
 2. Mettre à jour ses successeurs non encore marqués comme suit: $\pi(y) = \min (\pi(y), \pi(x) + L(x, y))$ tel que $y \in \Gamma^+(x) \cap (X - M)$
 3. Marquer le sommet x [$M = M \cup \{x\}$]

✿ PRINCIPE DE FONCTIONNEMENT.

💡 Graphe non orienté:

- 👉 Soit $\pi(i)$ la valeur de chaine du sommet ‘s’ vers le sommet ‘i’, on initialise: $\pi(s) = 0$ et $\pi(x) = \infty$ pour tout sommet $x \neq s$
- 👉 Soit M L’ensemble des sommets marqués, initialement il est vide ($M = \emptyset$)
- 👉 Tant qu’il existe un sommet non marqué ($M \neq X$) ou on n’est pas arrivé au sommet destinataire ($x \neq d$) faire :
 1. Choisir un sommet non marqué, soit x ($x \in X - M$), ayant le plus petit π [i.e. $\pi(x) = \min \{\pi(y) \text{ tq } y \in X - M\}$]
 2. Mettre à jour ses voisins non encore marqués comme suit: $\pi(y) = \min (\pi(y), \pi(x) + v(x, y))$ tel que $y \in \{z \in X \text{ tel que: } (x, z) \in E\} \cap (X - M)$
 3. Marquer le sommet x [$M = M \cup \{x\}$]

✿ PRINCIPE DE FONCTIONNEMENT.

💡 Exemple :(graphe non orienté)

- En utilisant l'algorithme de Dijkstra, trouver le PCC de «A» vers tous les autres sommets du graphe $G=(S,A,w)$,telle que :
 - 👉 $X=\{A,B,C,D,E,F\}$ est l'ensemble des sommets d'un graphe G.
 - 👉 $E=\{\{A,B\},\{A,C\},\{B,D\},\{B,E\},\{B,C\},\{C,E\},\{D,E\},\{D,F\},\{E,F\}\}$ est l'ensemble des arêtes entre les sommets adjacents d'un graphe G.

👉 La fonction de valuation:

$$V(A,B)=2$$

$$V(A,C)=10$$

$$V(B,D)=3$$

$$V(B,E)=1$$

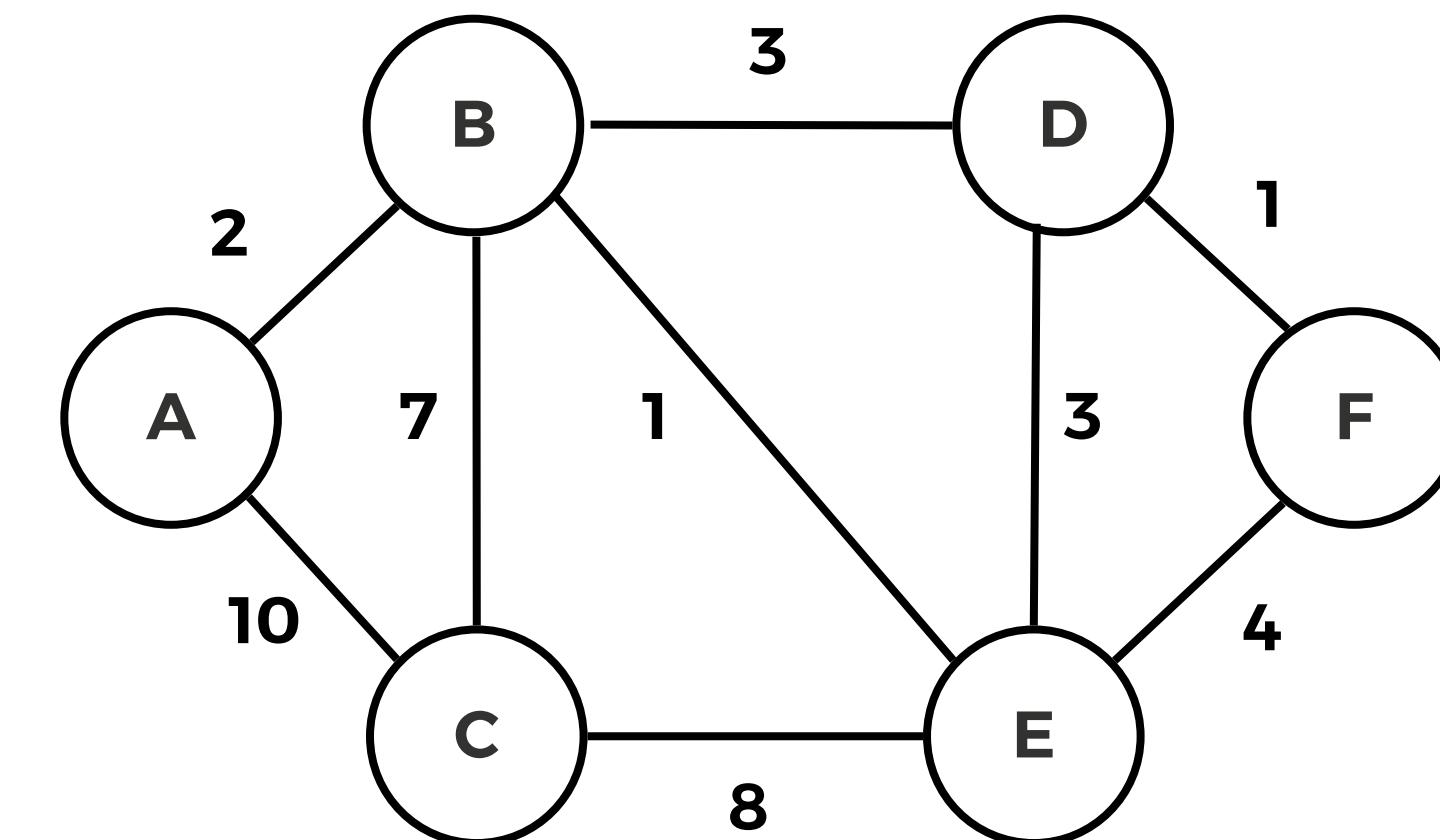
$$V(B,C)=7$$

$$V(C,E)=8$$

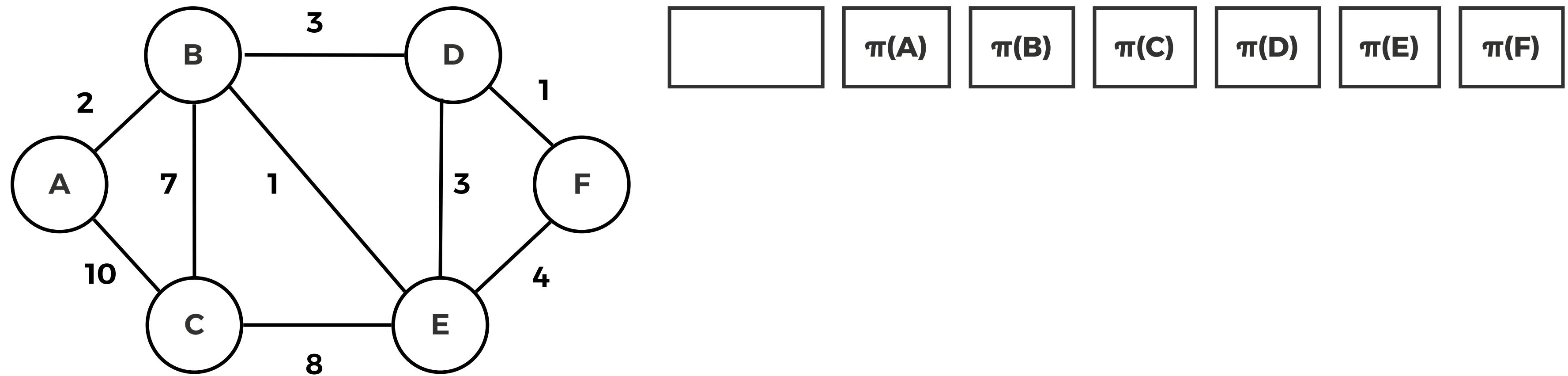
$$V(D,E)=3$$

$$V(D,F)=1$$

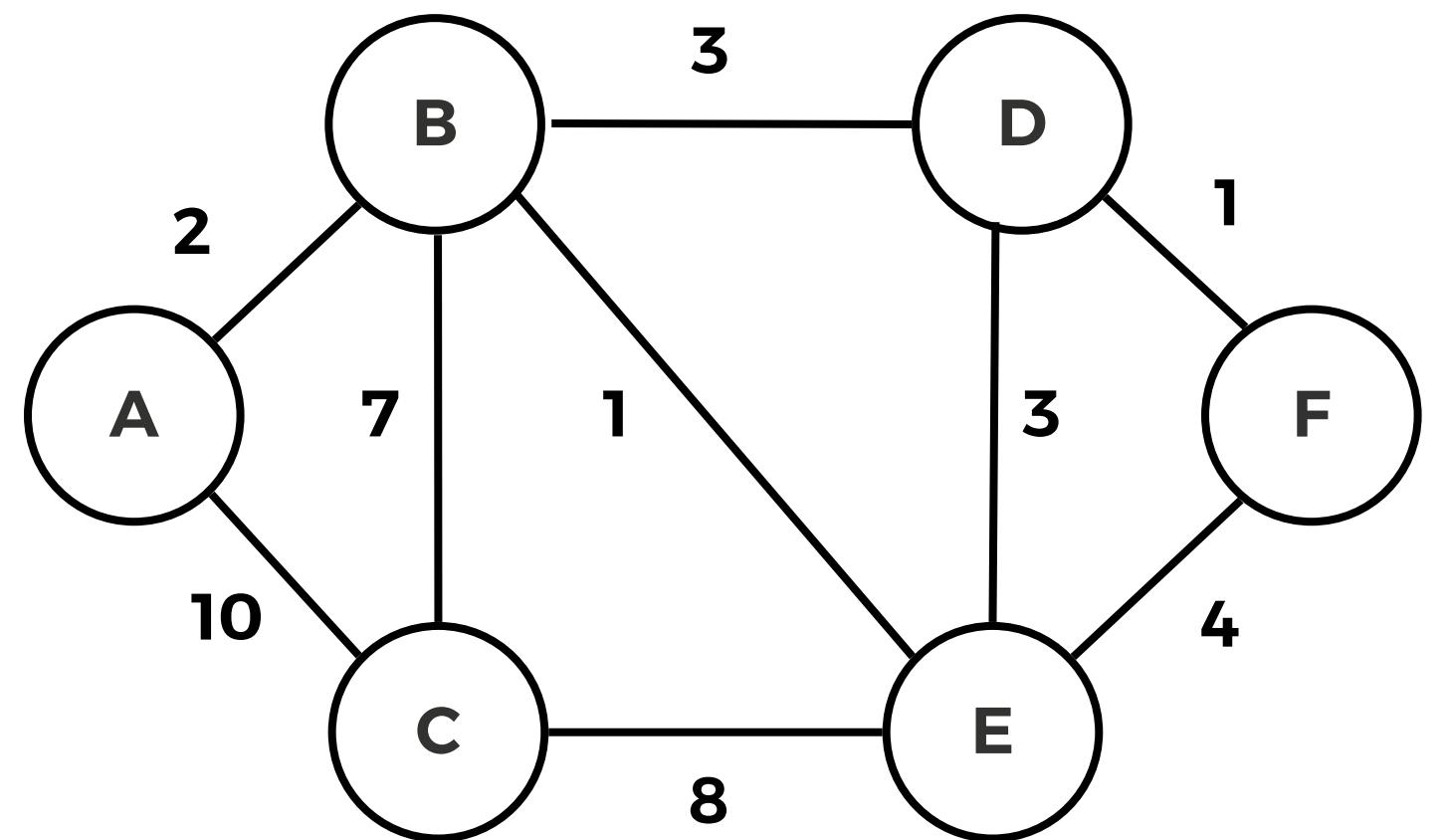
$$V(E,F)=4$$



✿ PRINCIPE DE FONCTIONNEMENT.

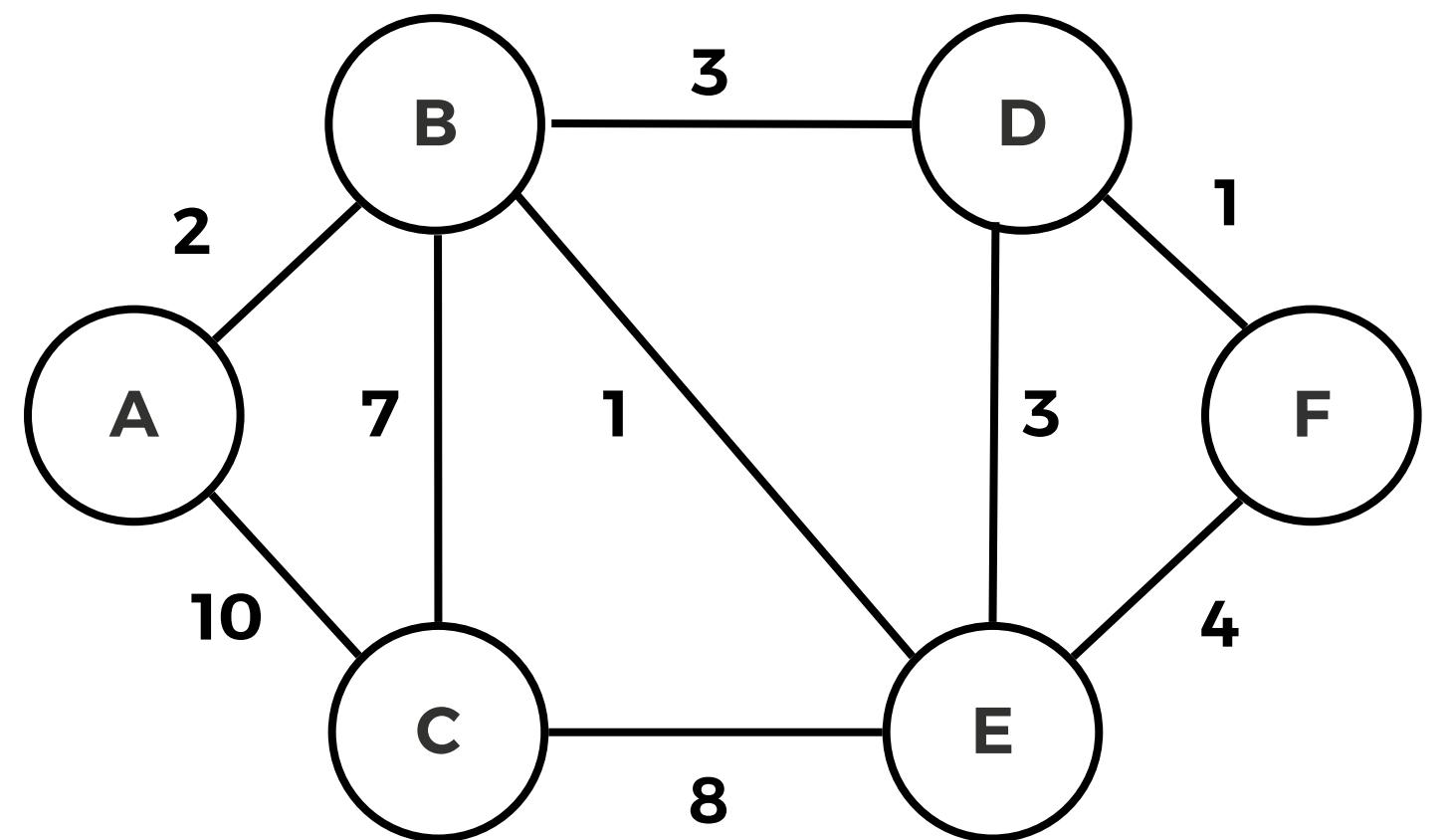


✿ PRINCIPE DE FONCTIONNEMENT.



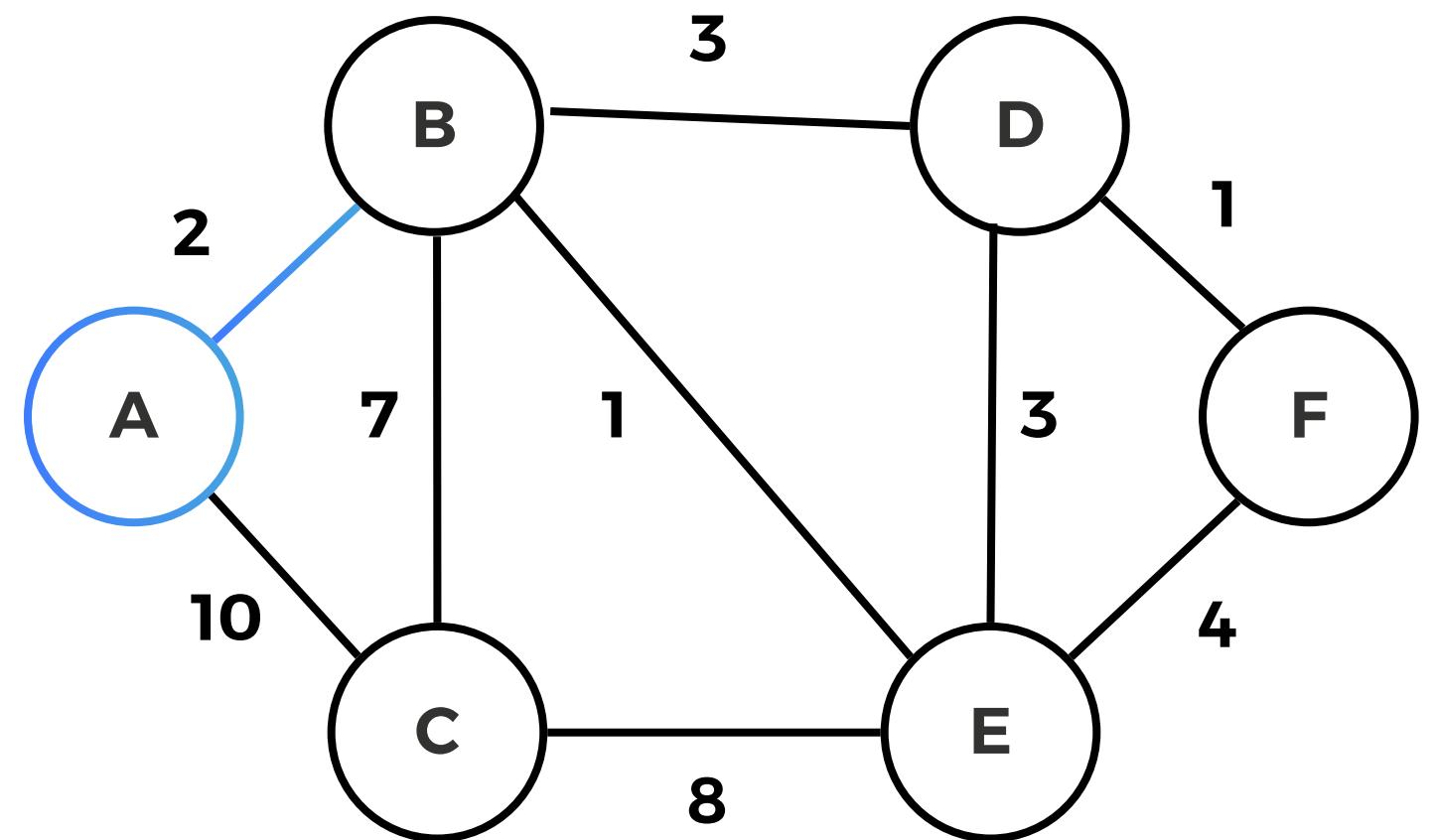
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞

✿ PRINCIPE DE FONCTIONNEMENT.



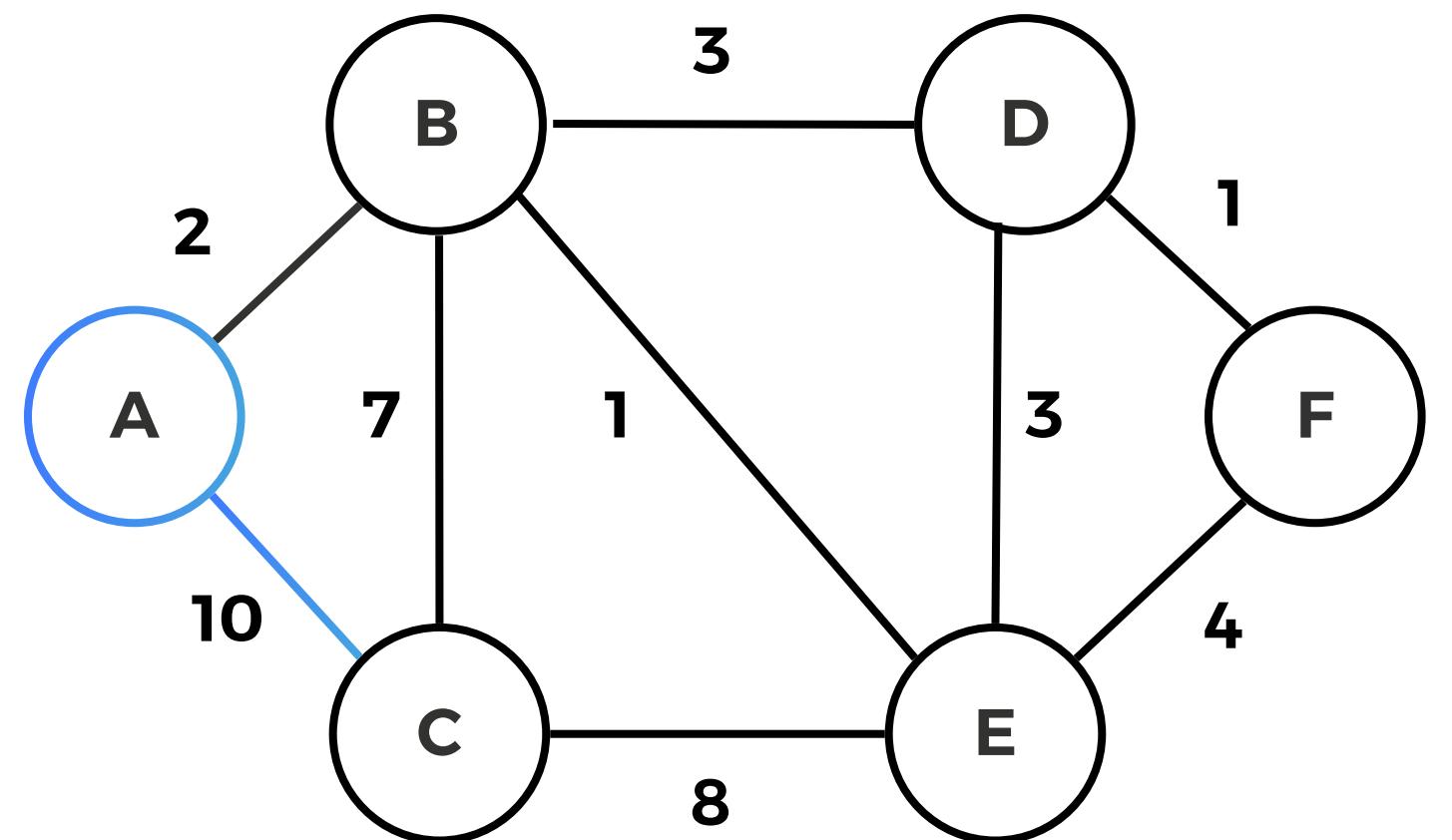
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞

✿ PRINCIPE DE FONCTIONNEMENT.



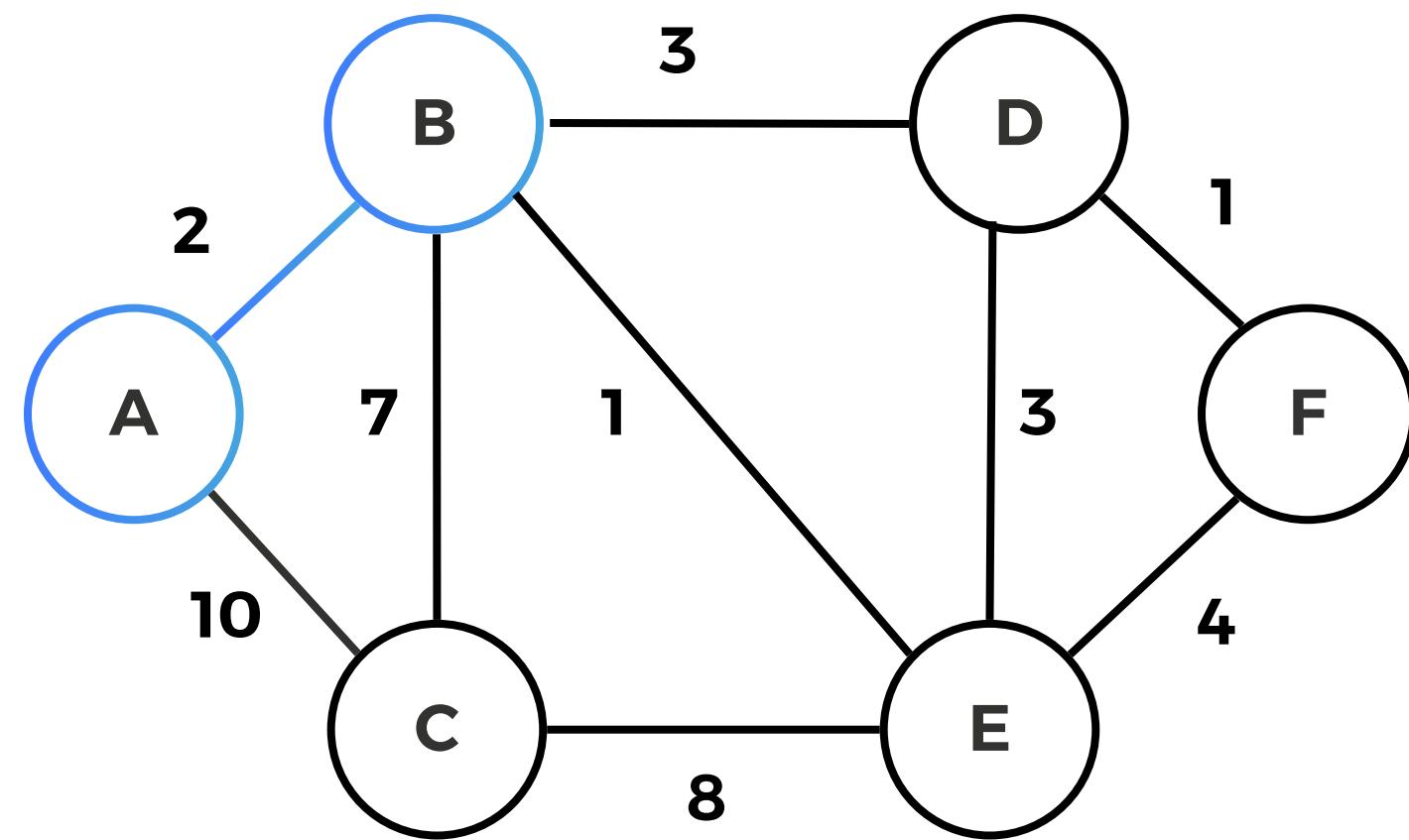
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)		0	∞	∞	∞	∞
1	1	0	2	∞	∞	∞

✿ PRINCIPE DE FONCTIONNEMENT.



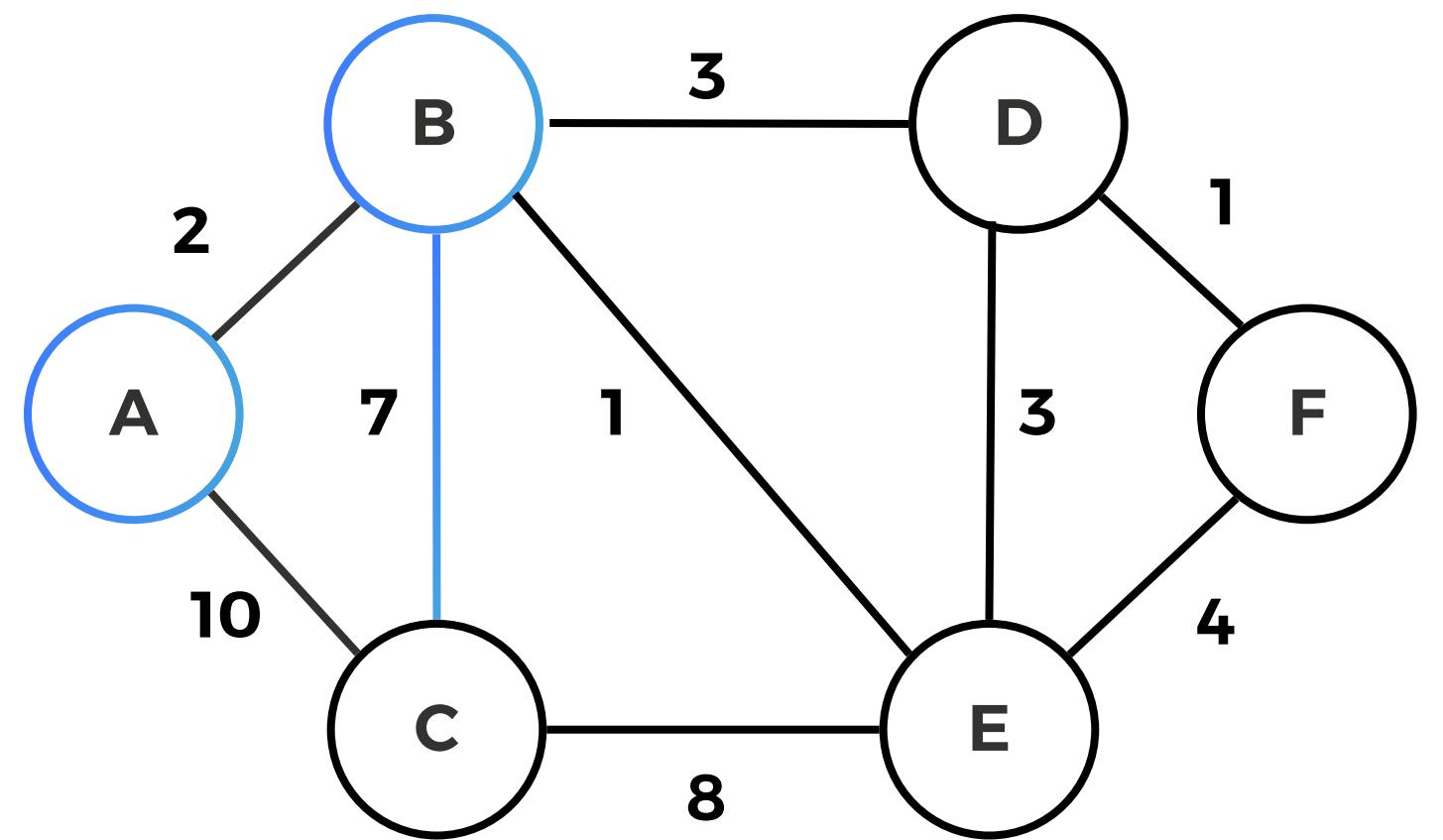
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	1	0	2	10	∞	∞

✿ PRINCIPE DE FONCTIONNEMENT.



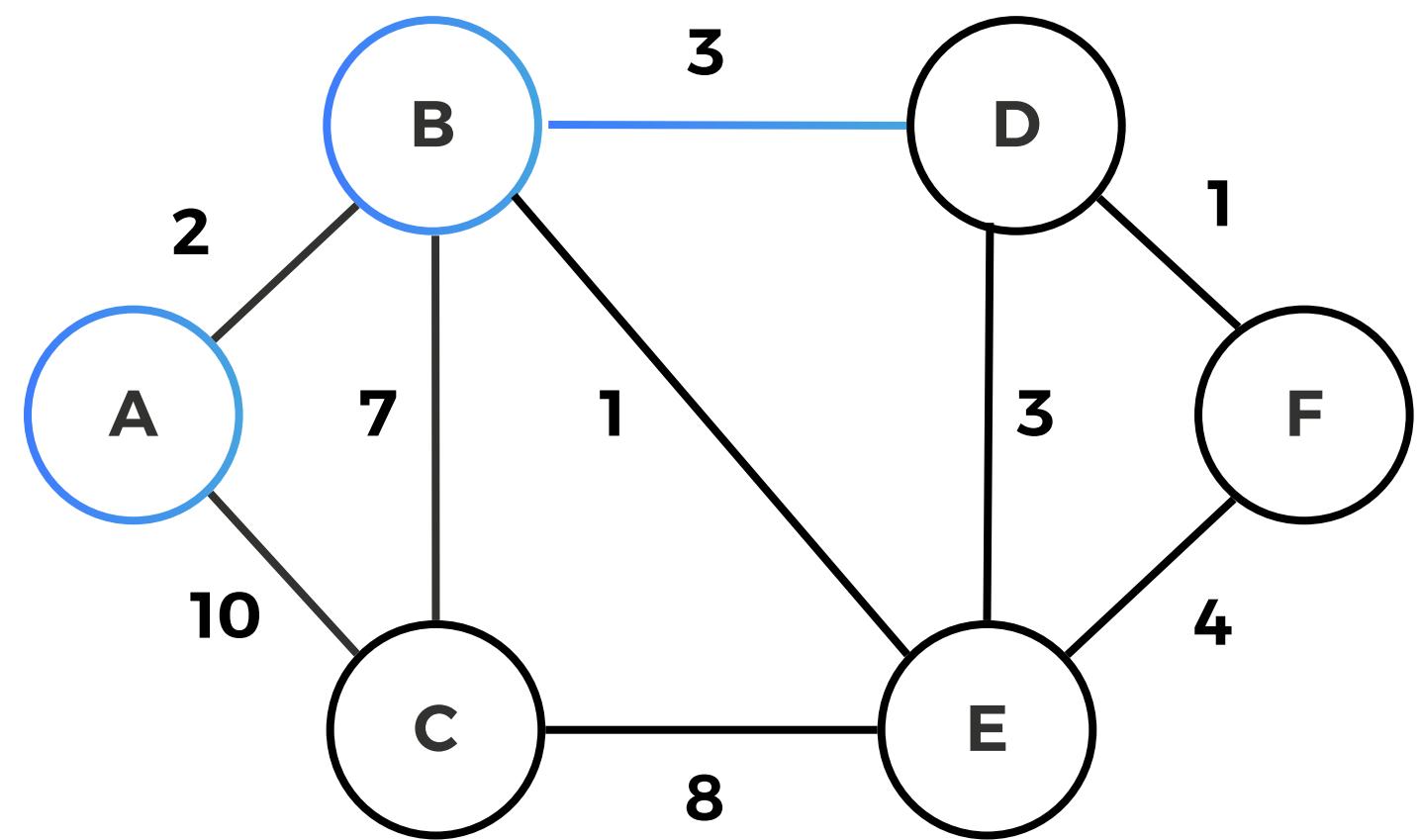
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0					
1		0	2	10	∞	∞
2			2	10	∞	∞
3						

✿ PRINCIPE DE FONCTIONNEMENT.



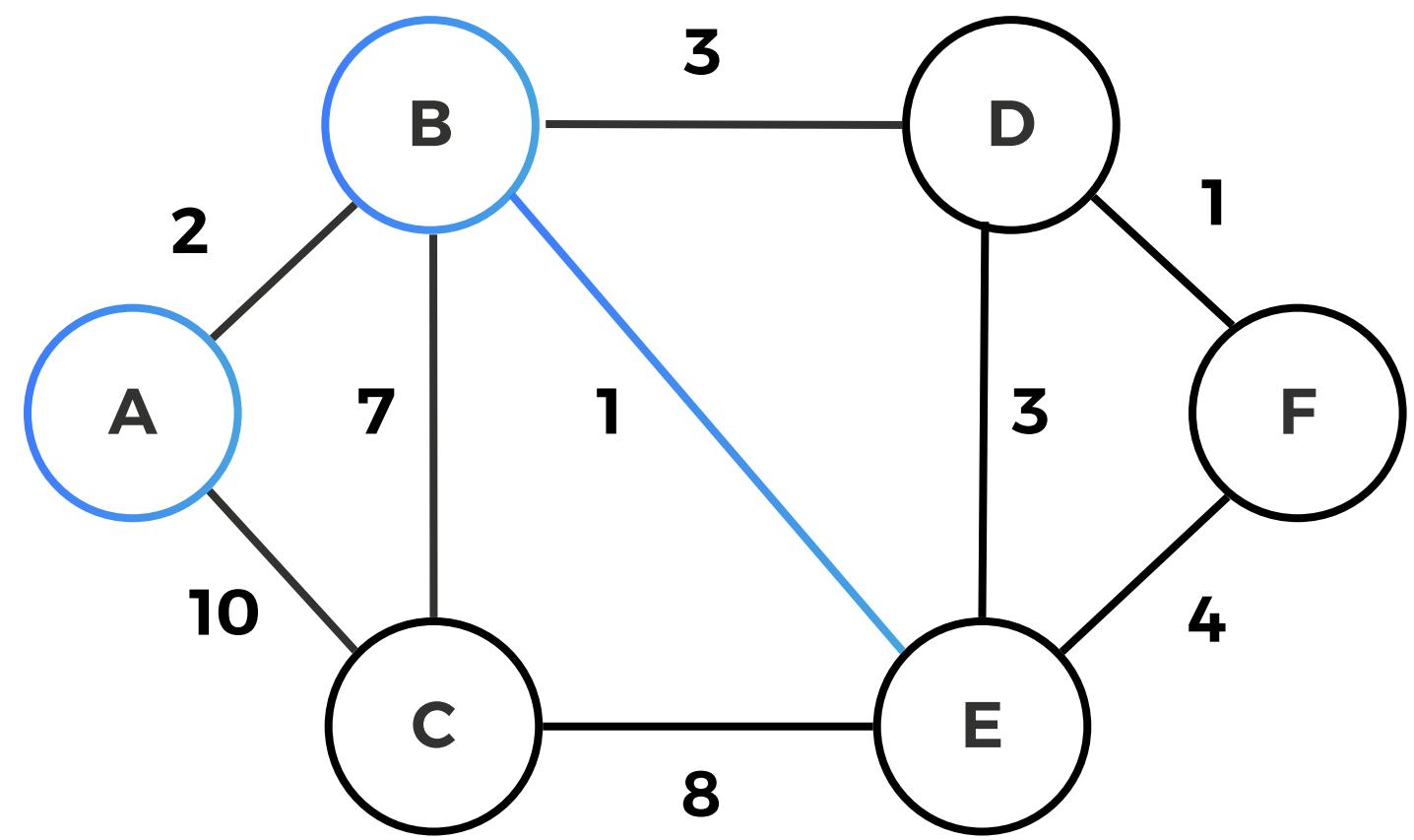
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0					
1		0	2	10		
2			2	9		
					∞	∞

✿ PRINCIPE DE FONCTIONNEMENT.



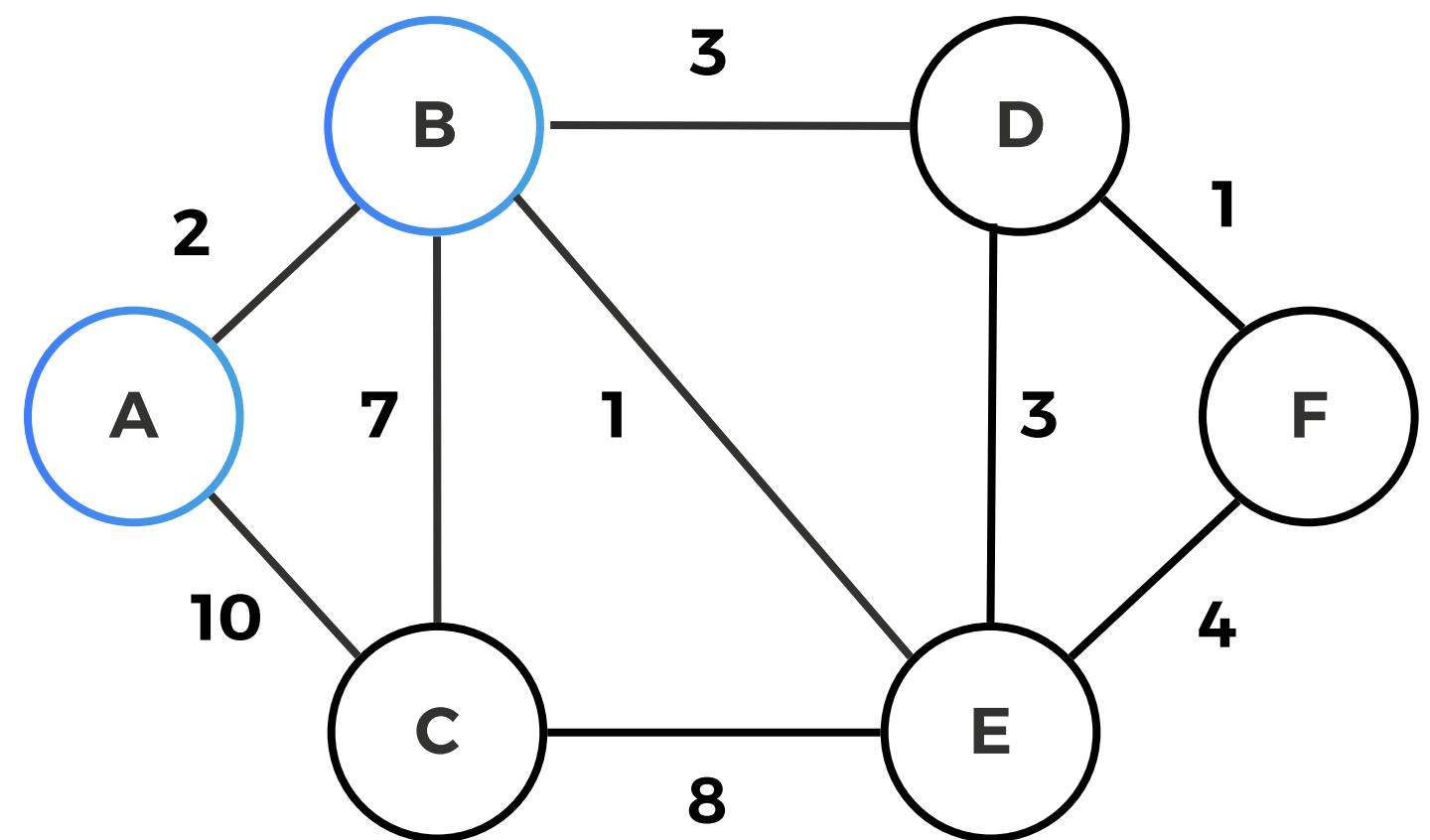
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0					
1		0	2	10	∞	∞
2			2	9	5	∞

PRINCIPE DE FONCTIONNEMENT.



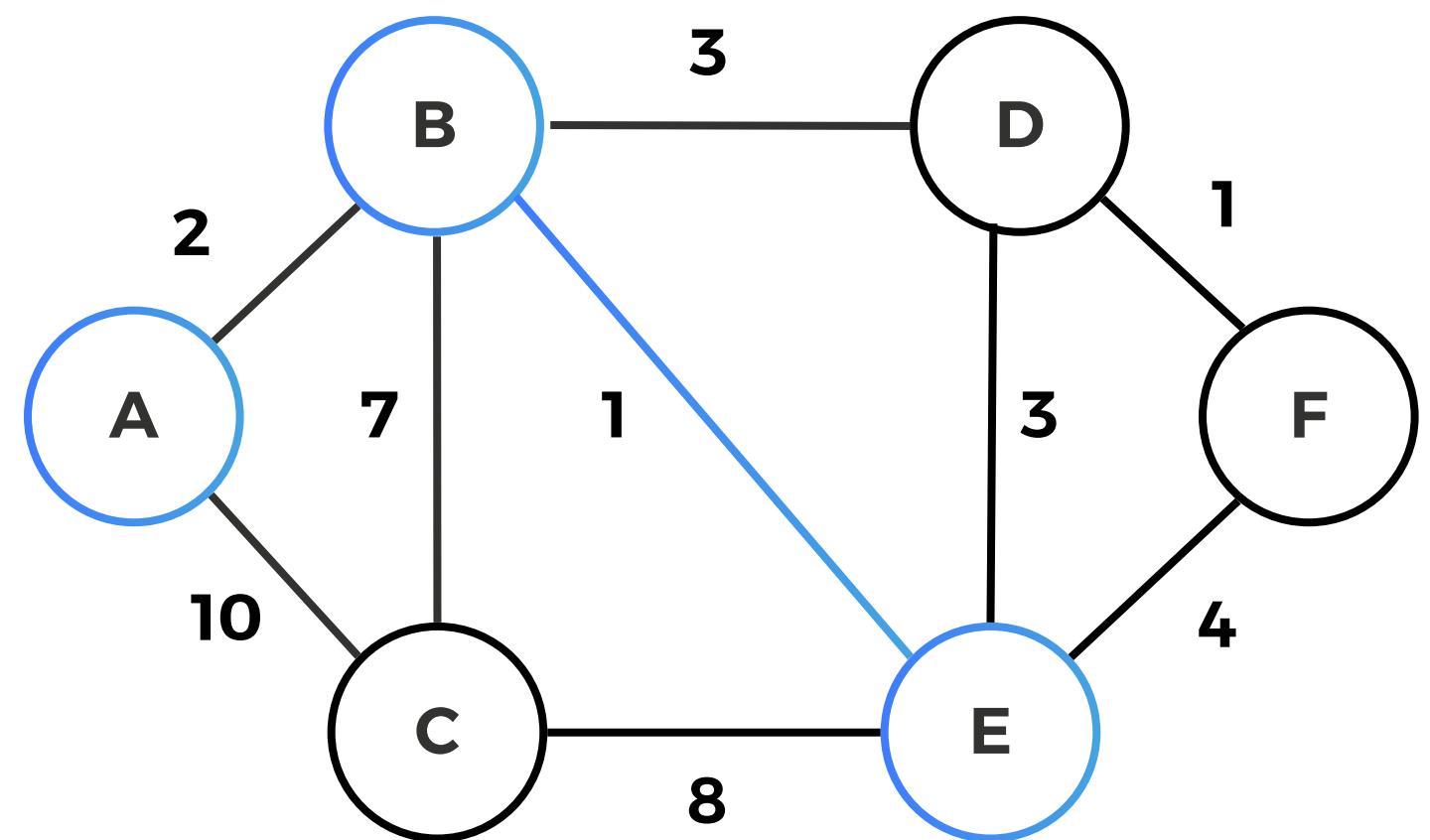
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞

✿ PRINCIPE DE FONCTIONNEMENT.



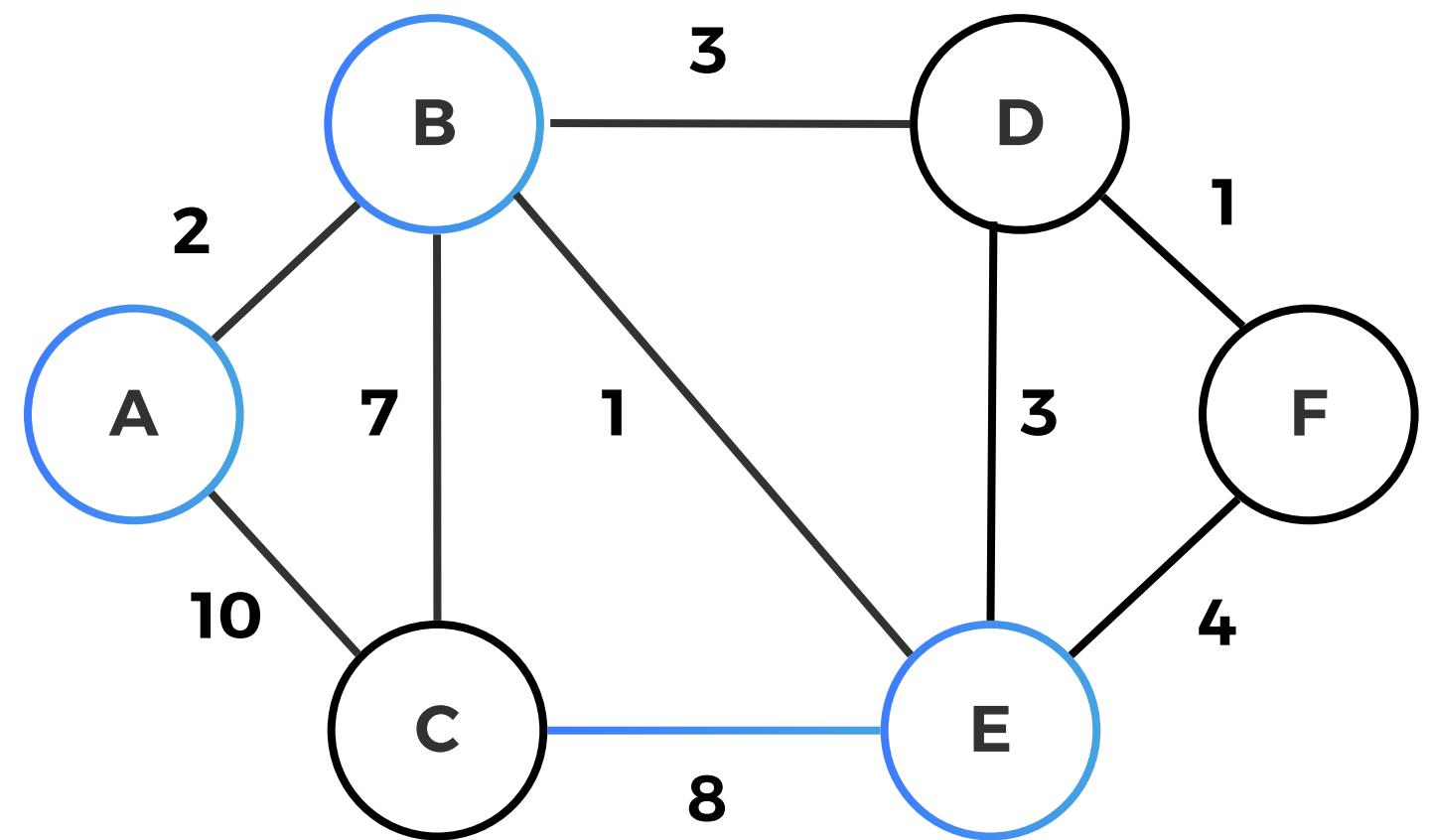
		$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)		0					
	1		0				
	2			2			
	3				9		
						5	
						3	
							∞

✿ PRINCIPE DE FONCTIONNEMENT.



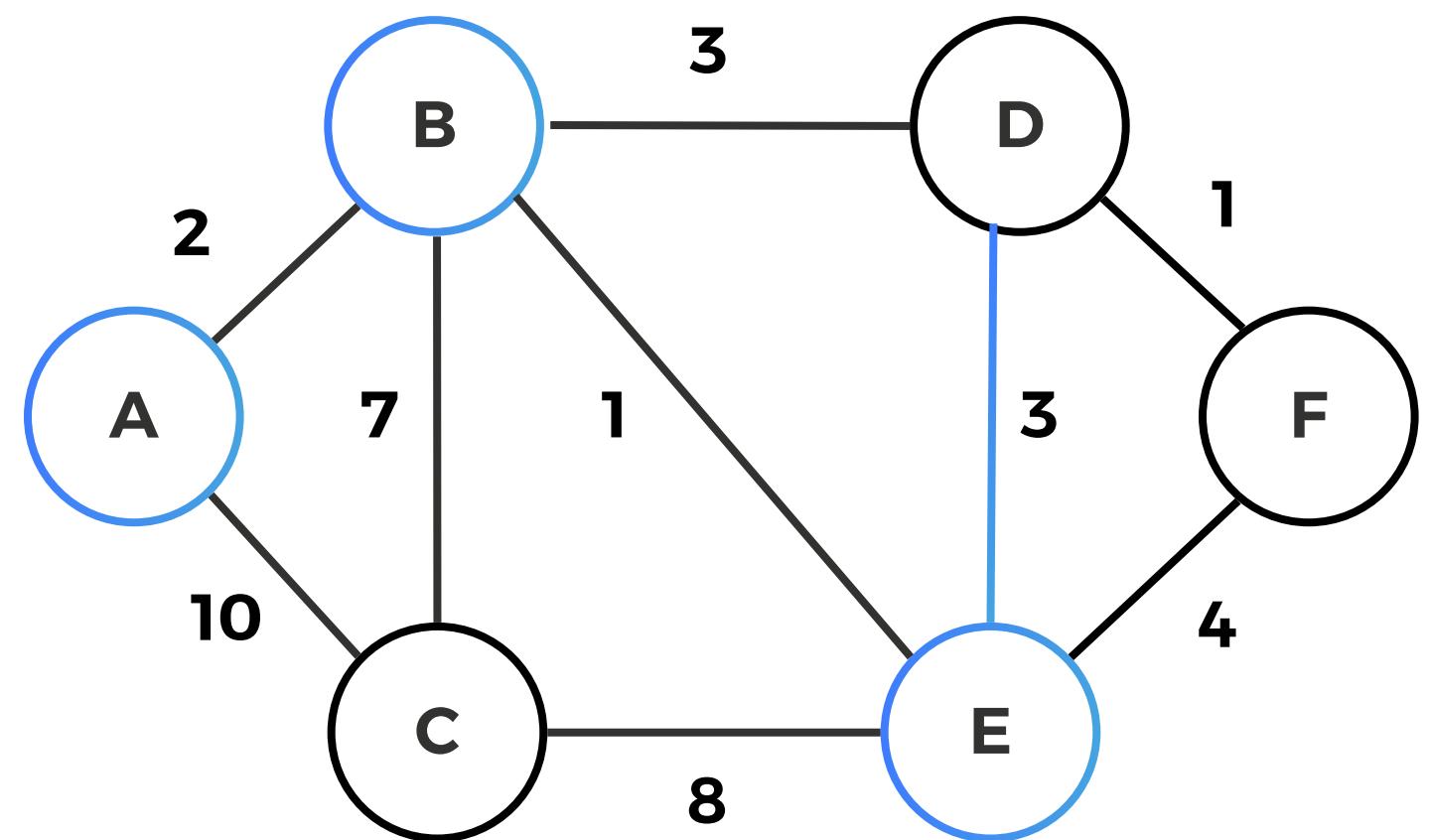
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	∞

✿ PRINCIPE DE FONCTIONNEMENT.



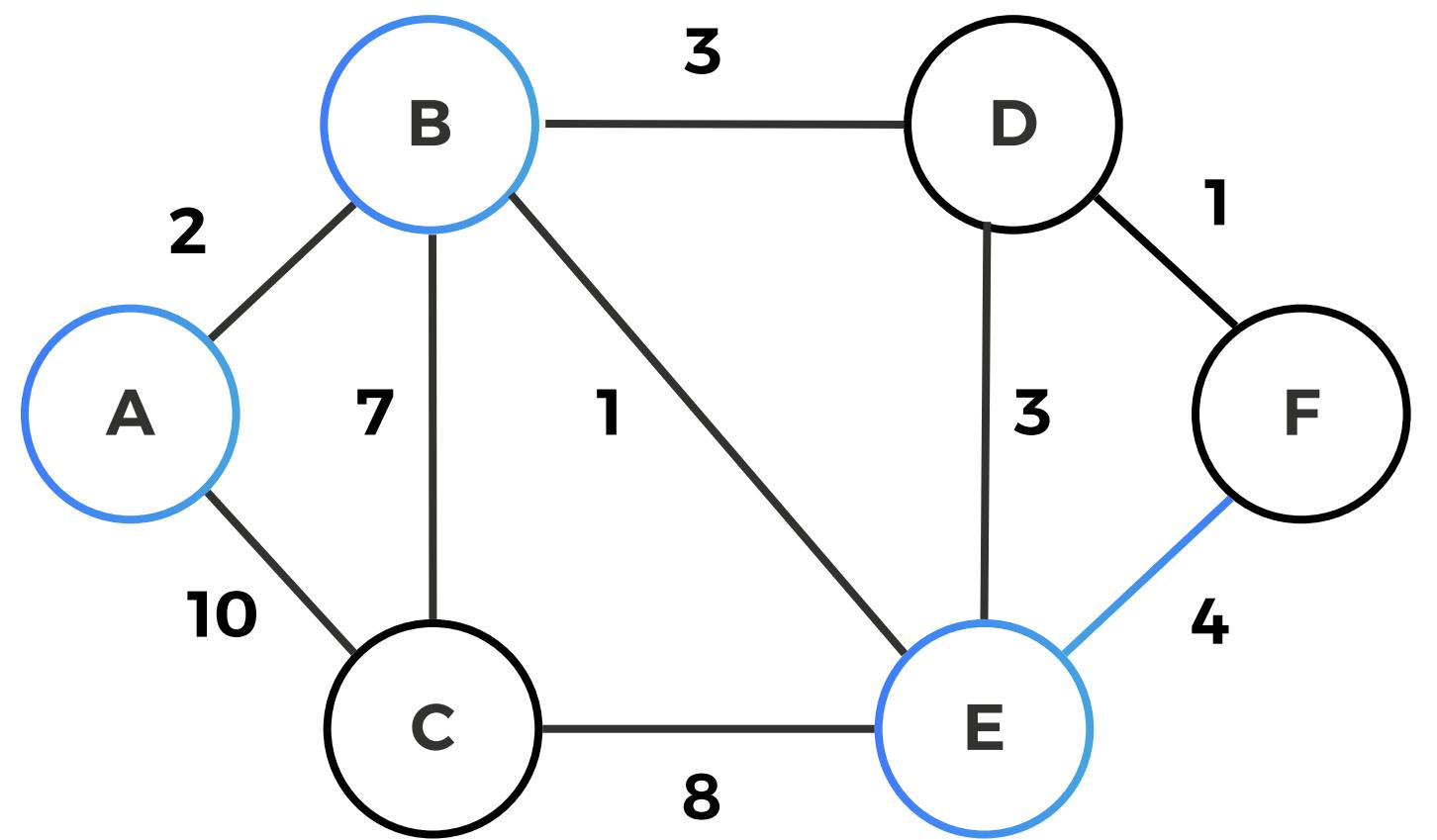
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2	2	2	9	5	3	∞
3			9	5	3	∞

✿ PRINCIPE DE FONCTIONNEMENT.



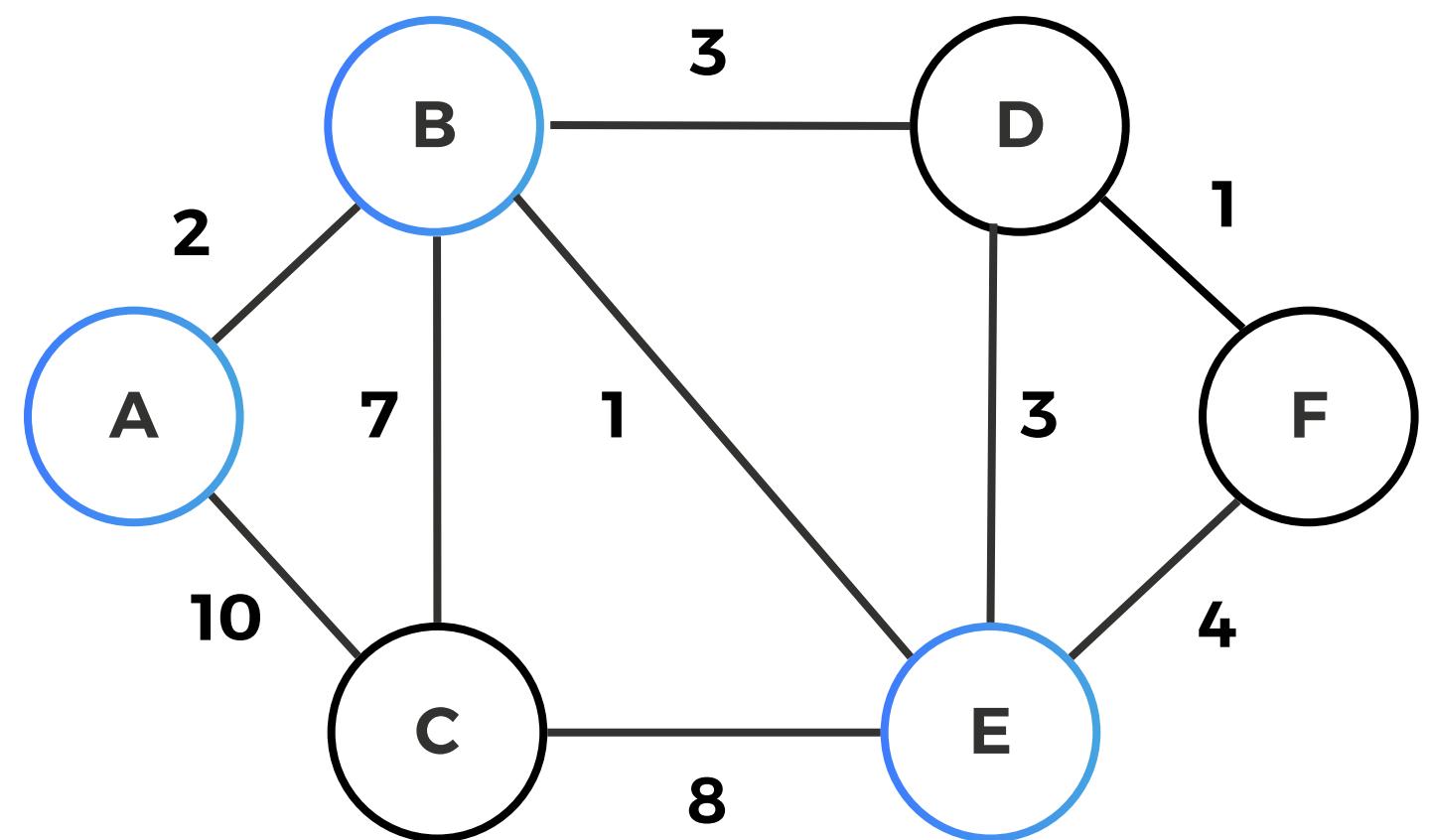
		$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)		0					
	1		∞	∞	∞	∞	∞
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
	10						

✿ PRINCIPE DE FONCTIONNEMENT.



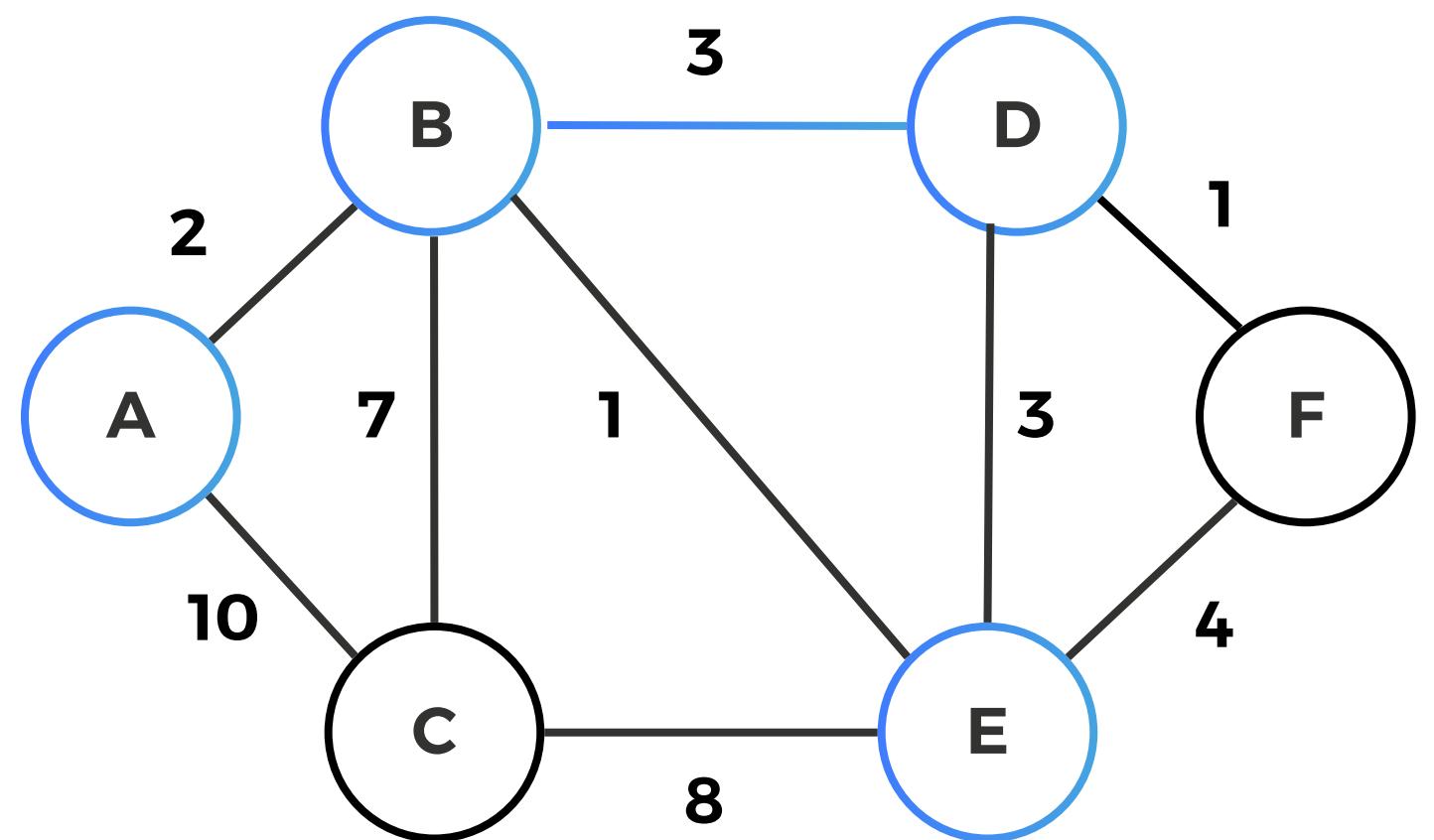
		$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
	0(init)	0					
		1	0	2	10	∞	∞
		2		2	9	5	3
		3			9	5	3
							7

✿ PRINCIPE DE FONCTIONNEMENT.



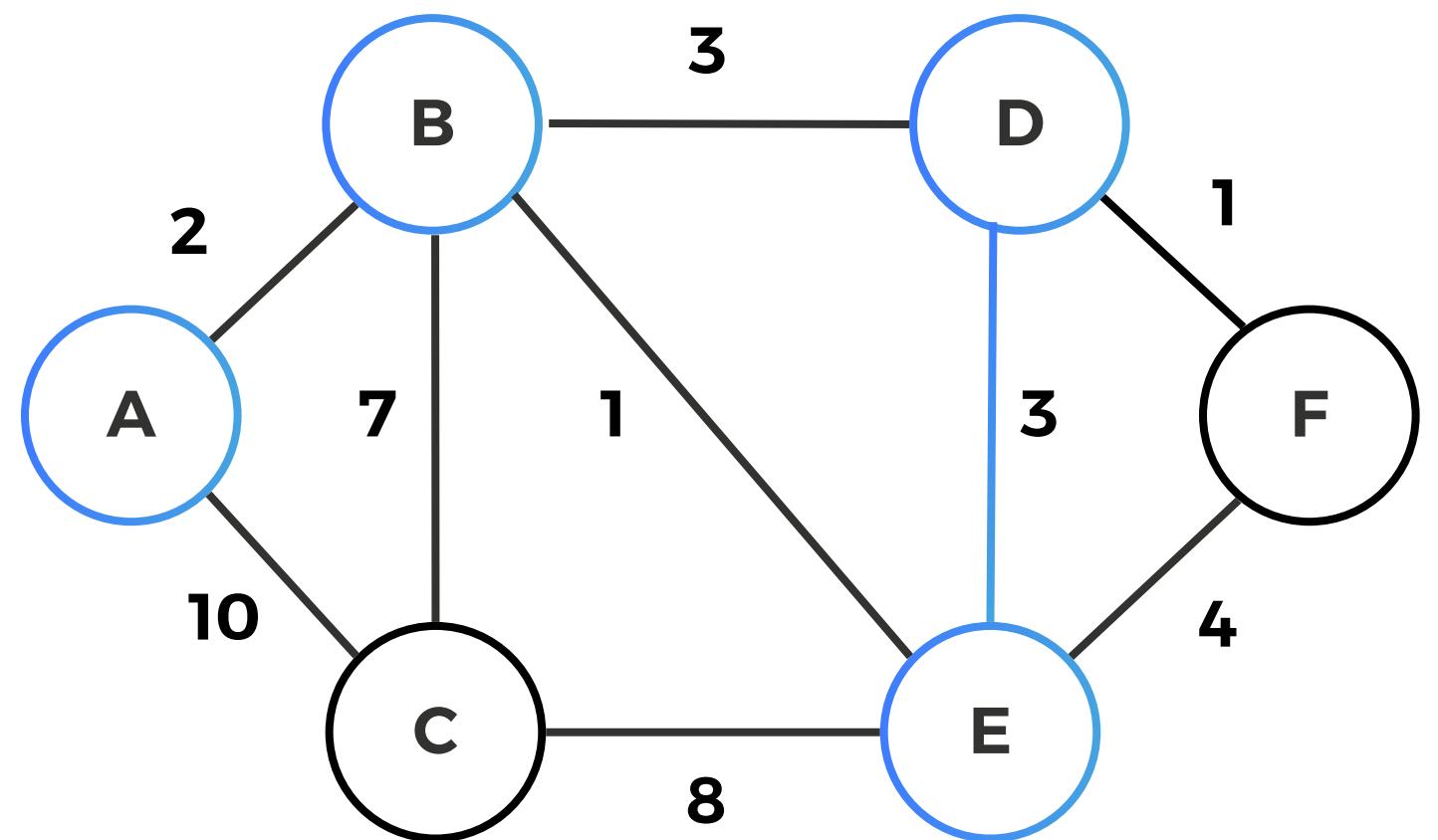
		$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)		0	∞	∞	∞	∞	∞
1		0	2	10	∞	∞	∞
2			2	9	5	3	∞
3				9	5	3	7
4					5		7

✿ PRINCIPE DE FONCTIONNEMENT.



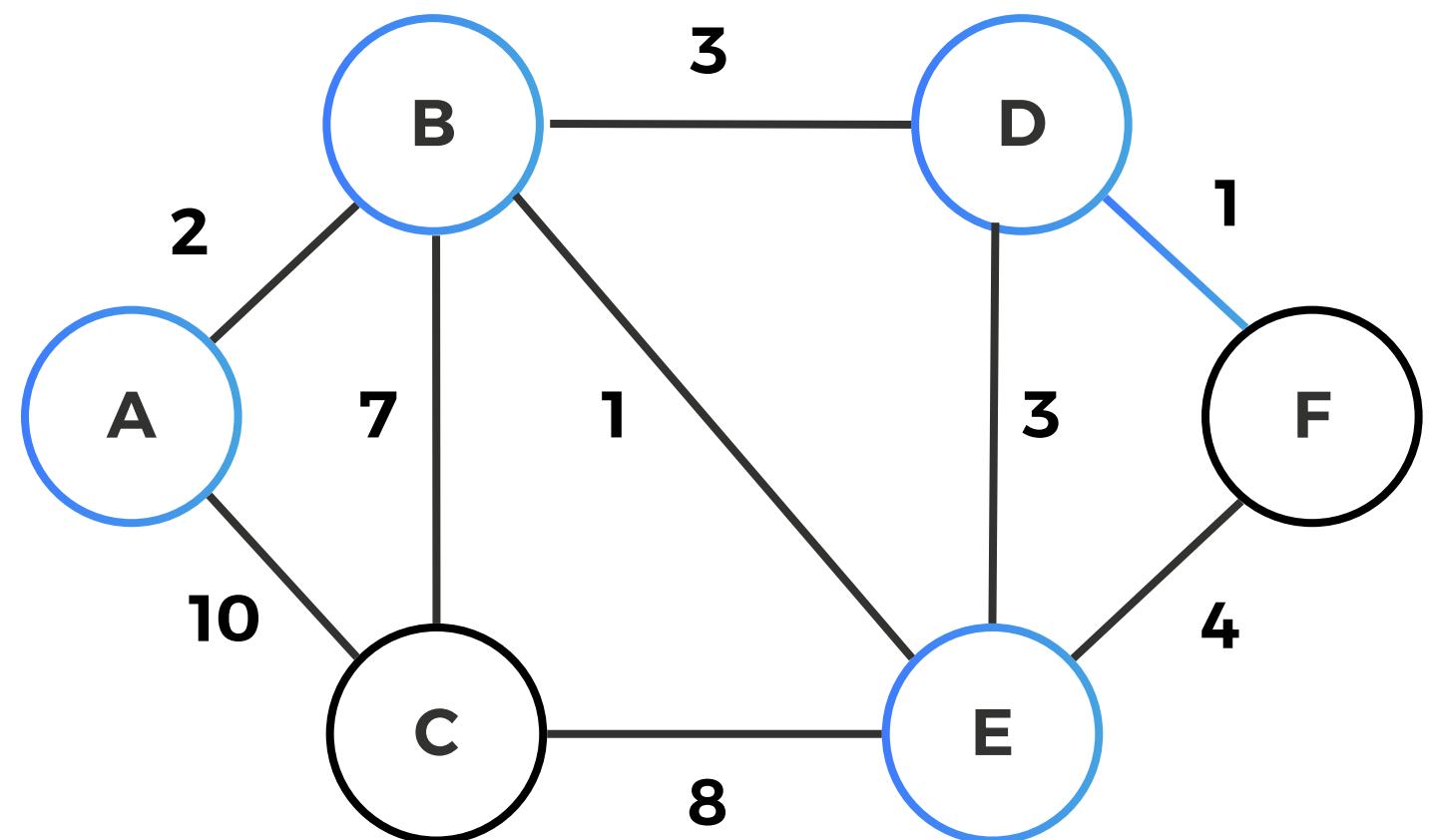
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$	
0(init)	0	∞	∞	∞	∞	∞	
1	0	2	10	∞	∞	∞	
2		2	9	5	3	∞	
3			9	5	3	7	
4				9	5		7

✿ PRINCIPE DE FONCTIONNEMENT.



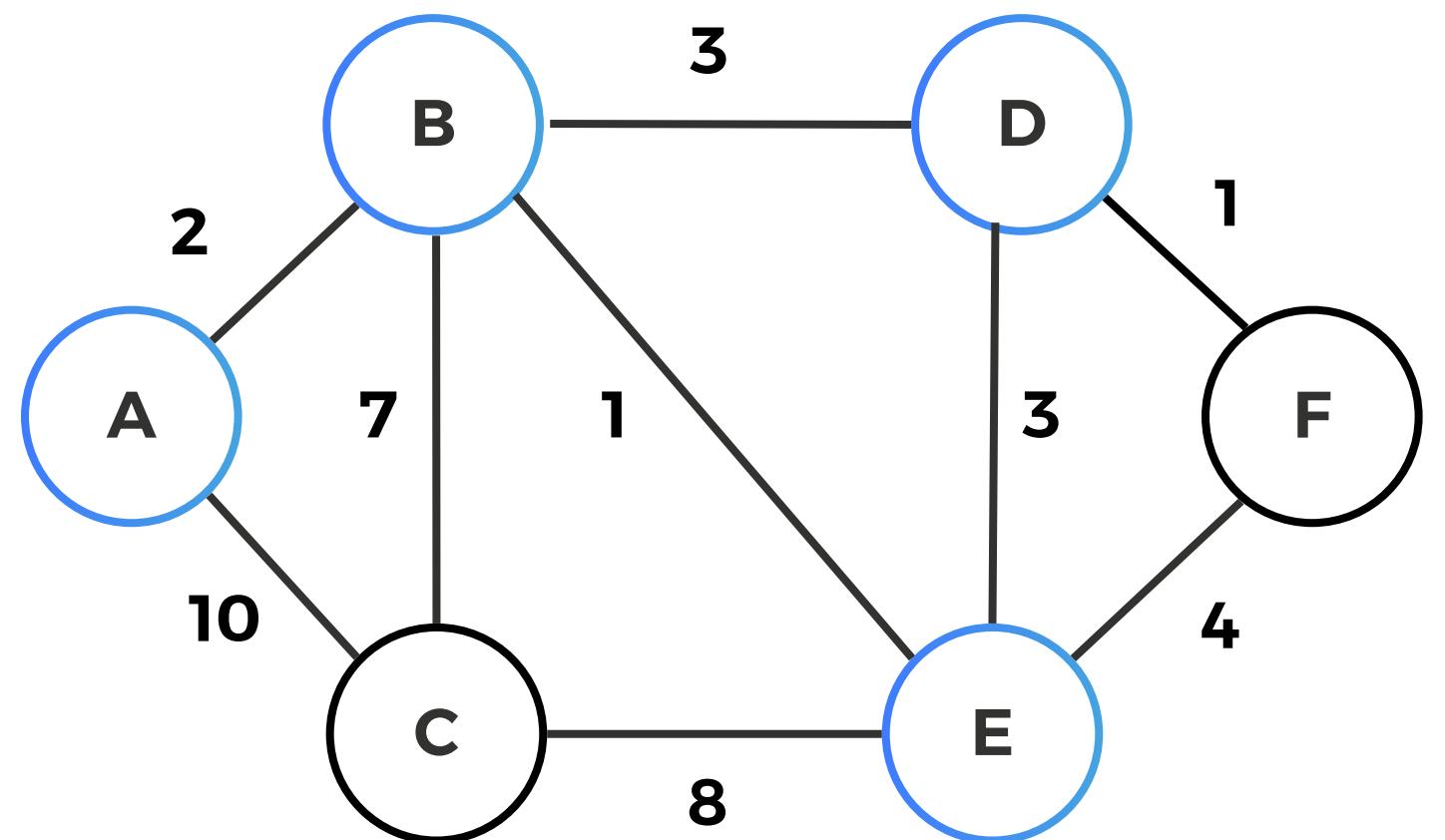
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	7

✿ PRINCIPE DE FONCTIONNEMENT.



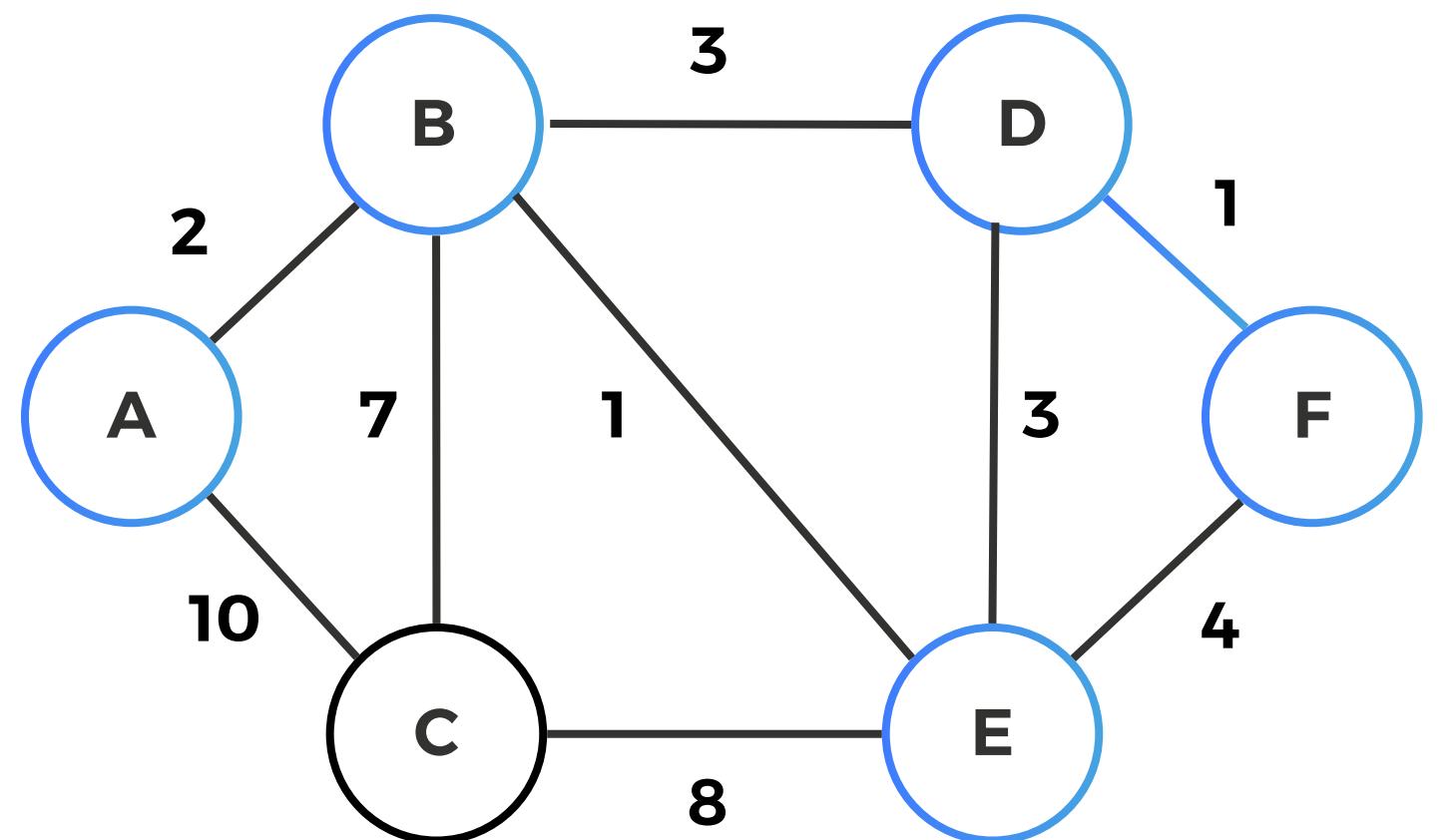
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	6

✿ PRINCIPE DE FONCTIONNEMENT.



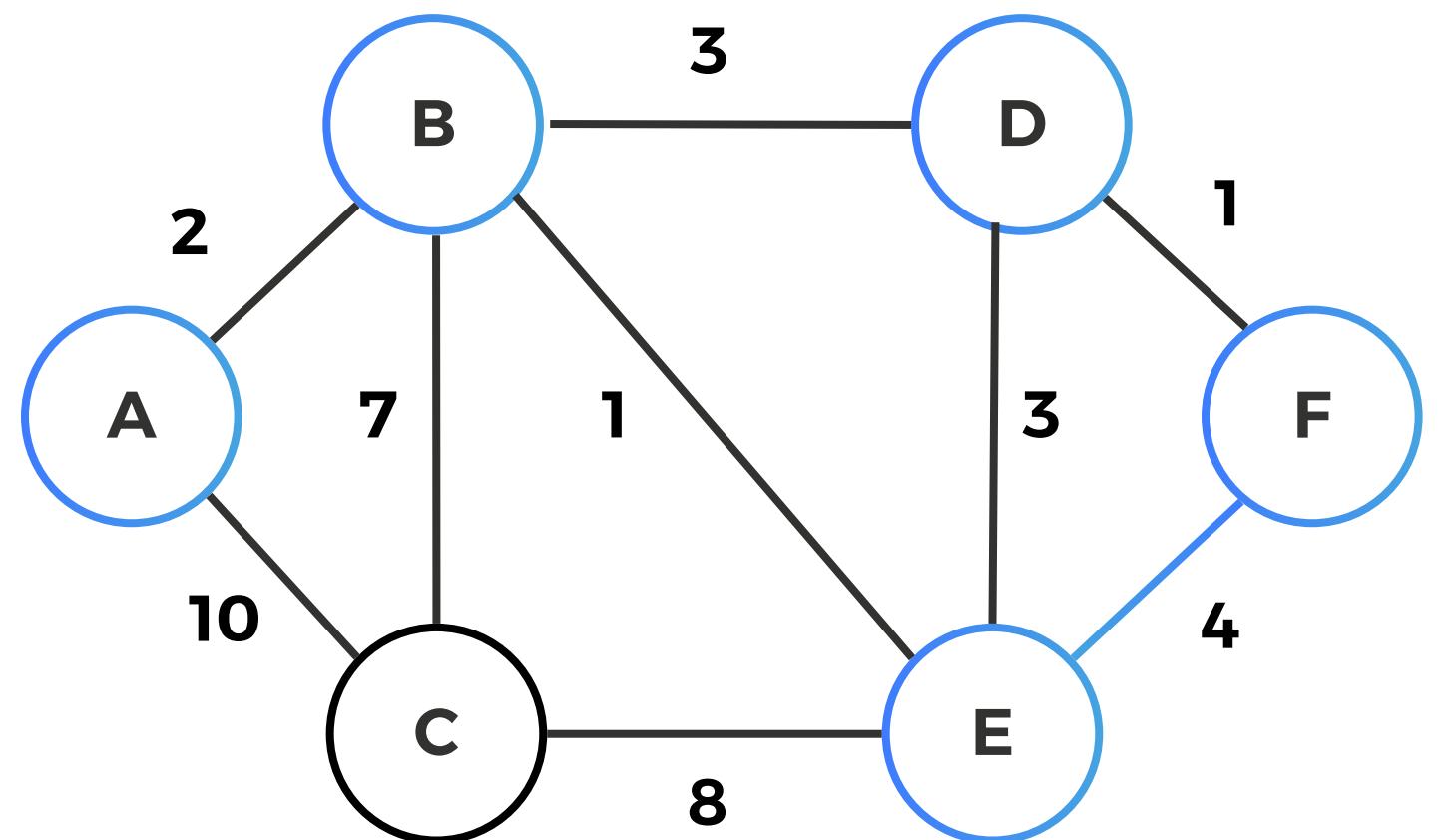
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2	2	2	9	5	3	∞
3			9	5	3	7
4				9	5	6
5				9		6

✿ PRINCIPE DE FONCTIONNEMENT.



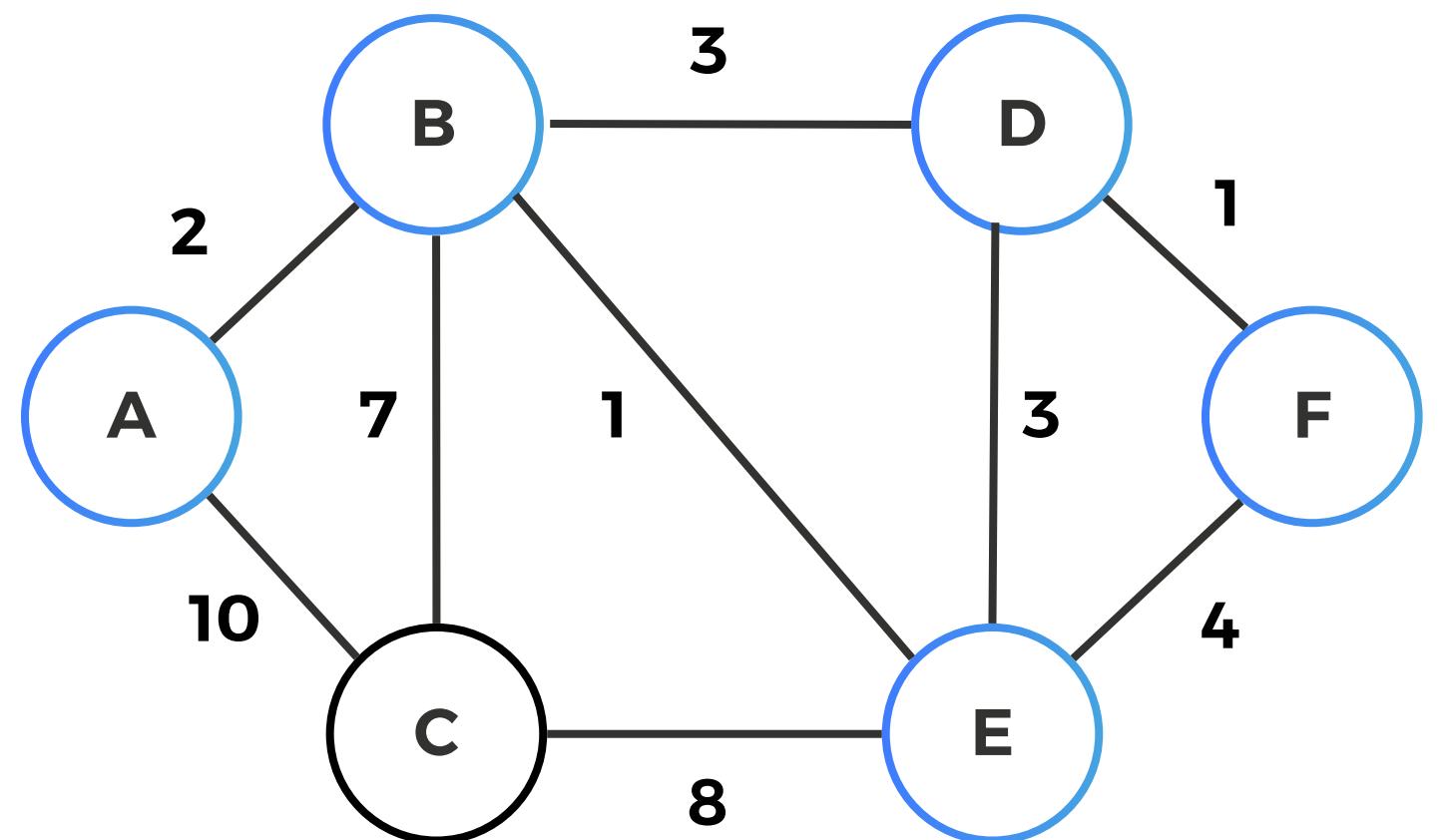
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	
5					9	
						6

✿ PRINCIPE DE FONCTIONNEMENT.



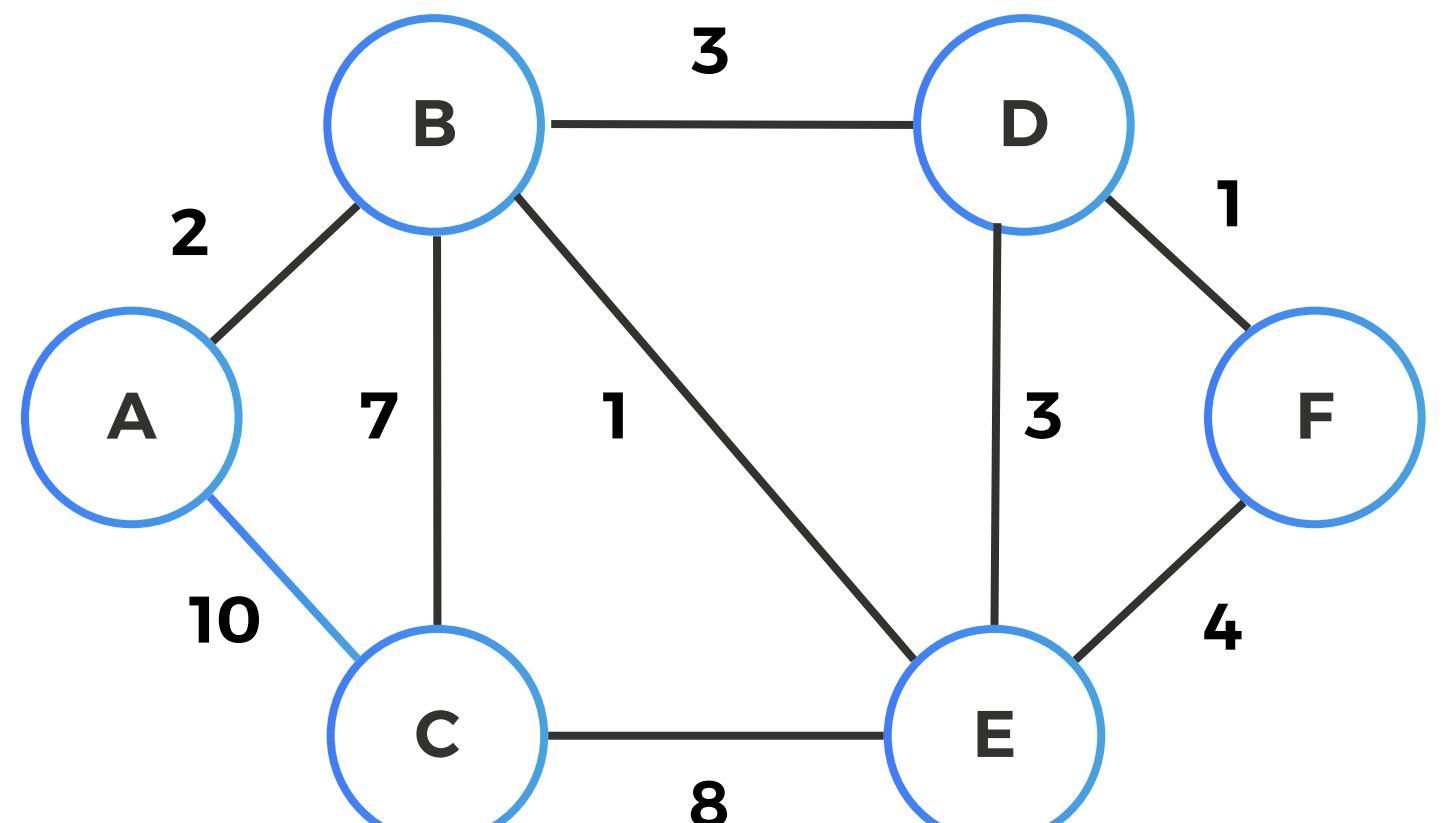
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	
5					9	
						6

✿ PRINCIPE DE FONCTIONNEMENT.



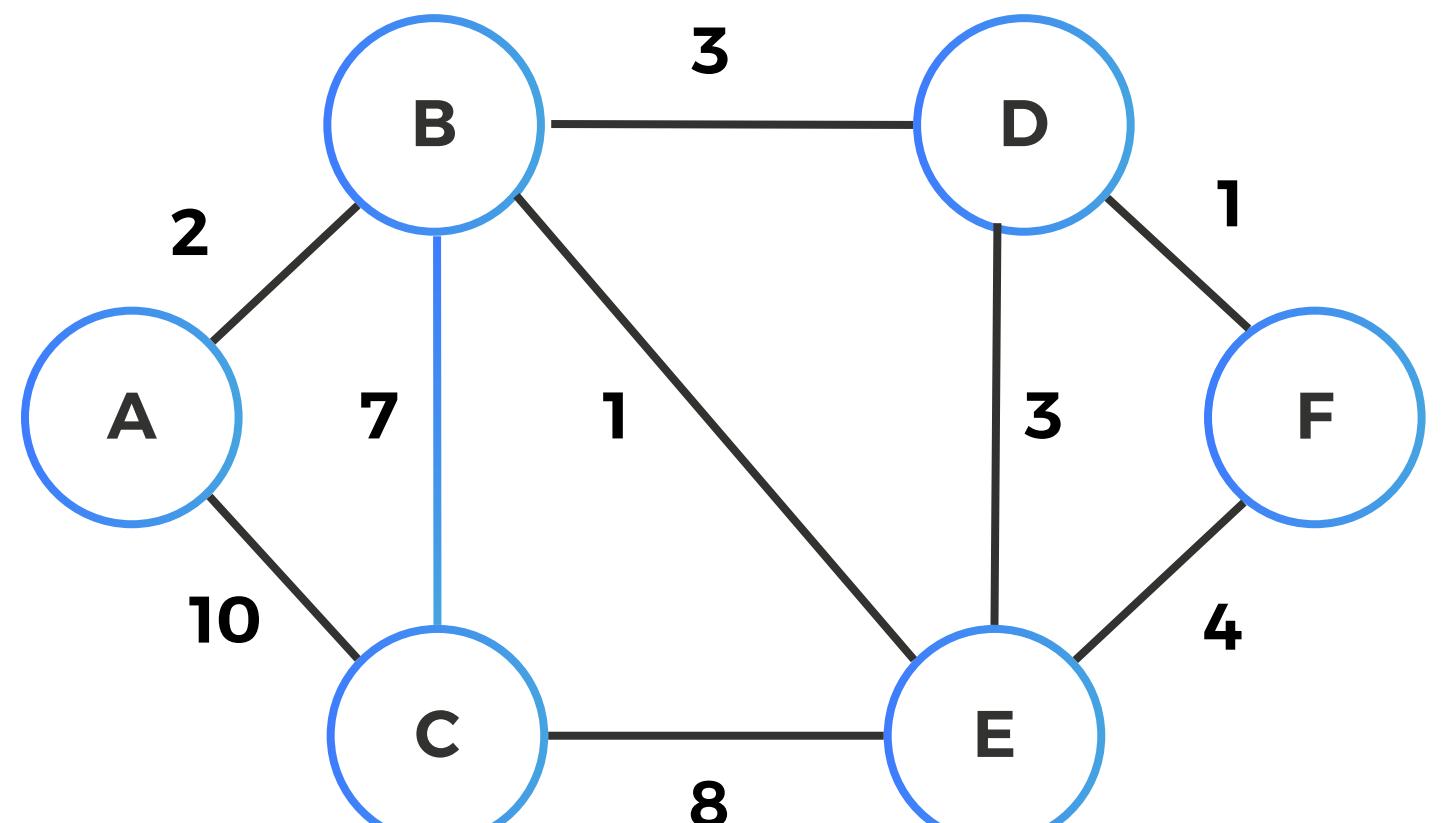
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				5		6
5				9		
6						6

✿ PRINCIPE DE FONCTIONNEMENT.



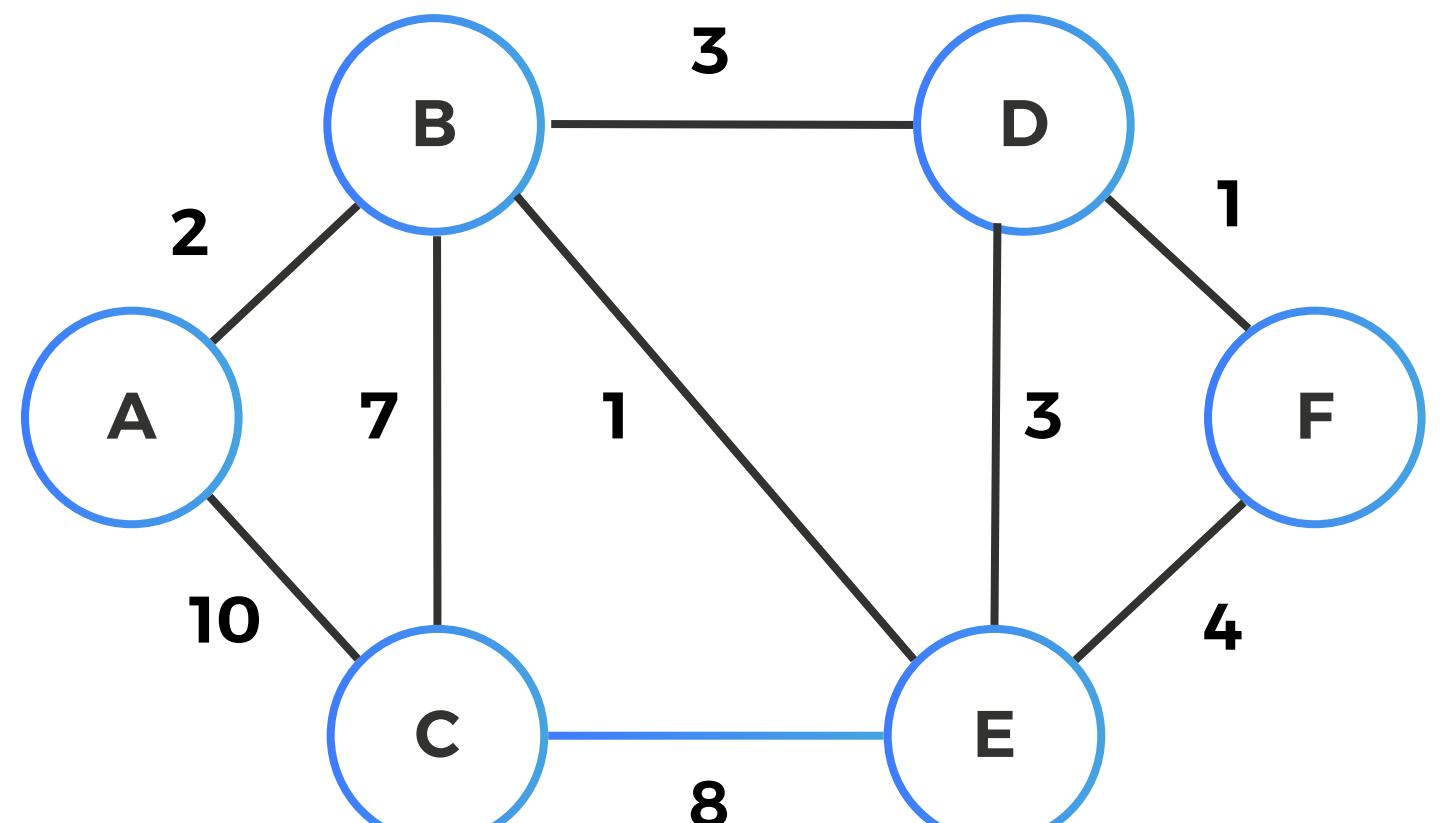
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	
5				9		
6						6

✿ PRINCIPE DE FONCTIONNEMENT.



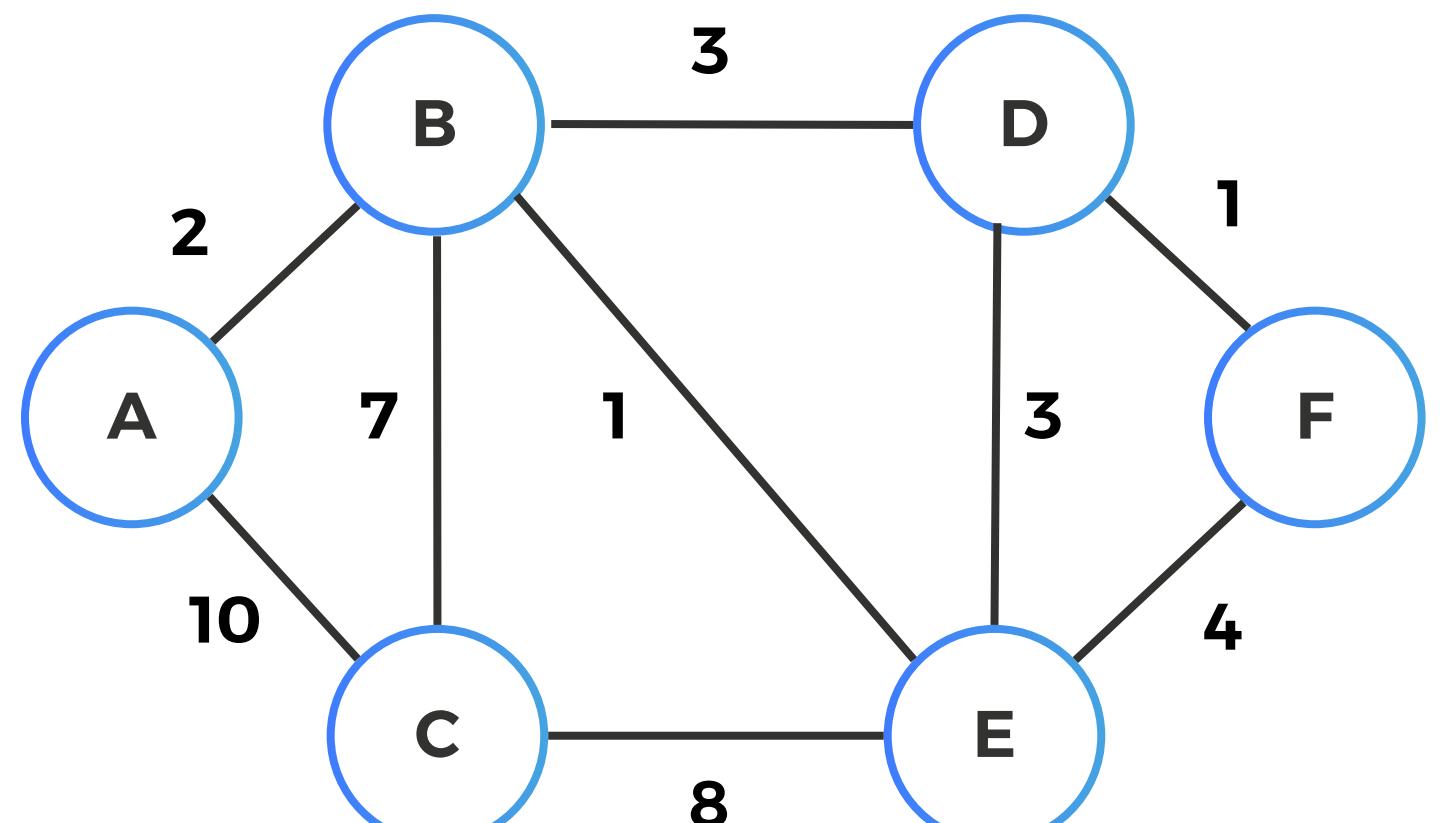
	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	
5				9		
6						6

✿ PRINCIPE DE FONCTIONNEMENT.



	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	
5				9		
6						6

✿ PRINCIPE DE FONCTIONNEMENT.



	$\pi(A)$	$\pi(B)$	$\pi(C)$	$\pi(D)$	$\pi(E)$	$\pi(F)$
0(init)	0	∞	∞	∞	∞	∞
1	0	2	10	∞	∞	∞
2		2	9	5	3	∞
3			9	5	3	7
4				9	5	
5				9		
6						6

✿ PRINCIPE DE FONCTIONNEMENT.

👉 Trouver un PCC allant du sommet « A » vers le sommet « F » dans un graphe G:

1. Garder les sommets dont la valeur du PCC est calculée.

Soit $G' = (X', E')$ le sous graphe induit engendré.

2. Grader les arêtes (i, j) du graphe G' qui vérifient la relation suivante:

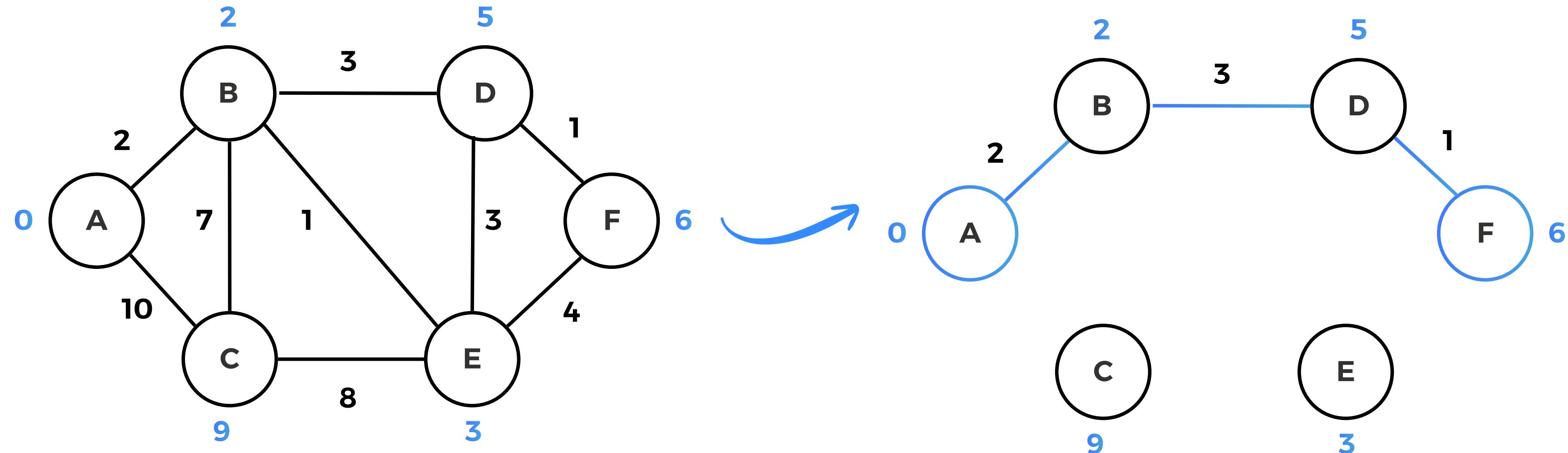
$v(i,j) = |\pi(j) - \pi(i)|$ ou tout simplement $\pi(i) + v(i,j) = \pi(j)$.

Soit $G'' = (X', E'')$ le graphe partiel trouvé.

3. Pour rechercher un PCC de « A » à un sommet donné « F », trouver une chaîne allant de sommet « s » au sommet « F » dans le graphe G'' .

✿ PRINCIPE DE FONCTIONNEMENT.

👉 Graphe partiel trouvé:



❖ IMPLEMENTATION

Entrées: $G = (X; U)$ un graphe avec une valuation positive c des arêtes, s un sommet;

Initialiser tous les sommets à non marqué ; Marquer s ;
 $L(s) \leftarrow 0$; // *Initialise le label de s à 0*

Tant que (il existe un sommet non marqué) **faire**

Pour chaque sommet y non marqué **faire**

Calculer $L(y) \leftarrow \min\{L(x) + c(x, y) \mid x \text{ voisin marqué de } y\}$;

Fin Pour

Choisir le sommet y non marqué de plus petit label L ;

Marquer y ;

Fait

✿ COMPLEXITÉ.

★ Dijkstra avec sélection linéaire (tableau):

La formule:

$$T_{DJ} = T_{int} + n * (T_{sommet\ min\ non\ marque} + T_{MAJ})$$

Alors:

$$T_{DJ} = O(2*n) + n * O(n)$$

$$T_{DJ} = O(n) + O(n^2)$$

Donc:

$$T_{DJ} = O(n^2)$$

Coloration des sommets

♣ INTRODUCTION.

- 👉 En théorie des graphes, le problème de coloration consiste en l'attribution de couleurs aux sommets (arêtes), de telle sorte que deux sommets adjacents (arêtes adjacentes) n'aient jamais la même couleur.
- 👉 Ce n'est qu'en 1976 que deux chercheurs américains, K. Appel et W. Haken, ont pu répondre affirmativement à cette conjecture des quatre couleurs. La carte à colorer a été remplacée par un graphe, chaque pays étant représenté par un sommet et deux pays voisins étant reliés par une arête.
- 👉 La coloration des graphes intervient dans différents problèmes: les problèmes d'optimisation avec contraintes d'incompatibilités comme le problème de planning des examens/événements, le problèmes de transport de produits chimiques.

❖ DÉFINITION.

👉 Coloration des sommets:

- Actuellement, les problèmes de coloration qui ont été étudiés le plus souvent sont ceux qui s'intéressent à la coloration de sommets.
- Étant donné un graphe simple non orienté $G = (V, E)$, une coloration propre des sommets consiste à affecter à chaque sommet de ce graphe une couleur de telle sorte que deux sommets adjacents n'aient pas la même couleur.
- Une k -coloration propre des sommets d'un graphe $G = (V, E)$ est une fonction $c : v \rightarrow c(v)$ associant à tout sommet $v \in V$ une couleur $c(v)$, avec $c(v)$ un entier naturel entre 1 et $|V|$, en s'assurant que $c(v) \neq c(u)$ pour toute arête $[u, v] \in E$.

❖ ALGORITHME.

pour trouver la coloration des sommets dans graphe(orienté ou non),nous allons voir l'algorithme suivant :

👉 Algorithme de DSATUR:

Cet algorithme consiste a colorer séquentiellement le graphe en visitant les sommets par ordre de degré de saturation qui est définit dynamiquement.

✿ PRINCIPE DE FONCTIONNEMENT.

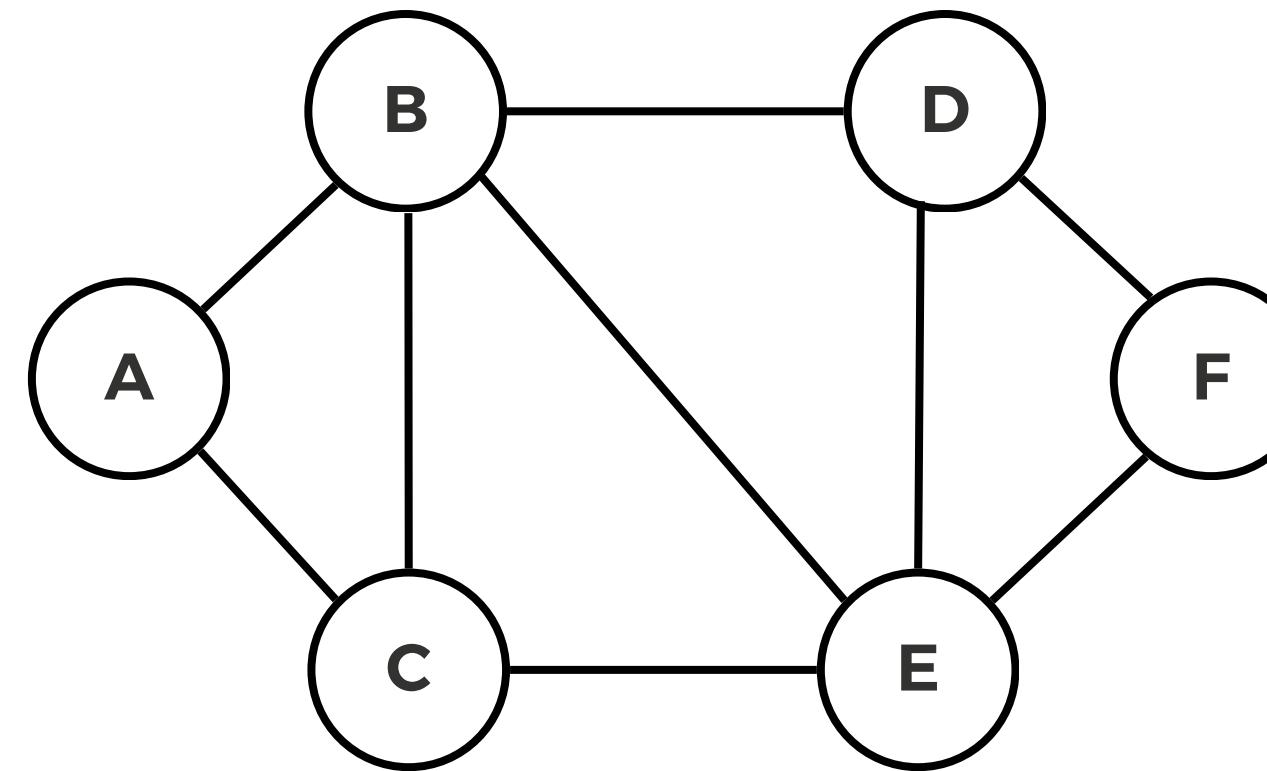
💡 Graphe orienté:

- 👉 Trier les sommets par ordre décroissant de leur degré
- 👉 Initialiser le DSAT de chaque sommet comme suit: $DSAT(i) = \text{degré}(i)$
- 👉 Tant qu' il existe un sommet non colorié:
 - a. Choisir un sommet avec DSAT maximum, en cas d'égalité celui de plus grand degré.
 - b. Colorier ce sommet avec la première couleur possible
 - c. Mettre à jour DSAT pour les sommets adjacents a ce sommet:
 $DSAT(i) = \text{nombre de couleurs différentes utilisées par les sommets adjacents au sommet } i$

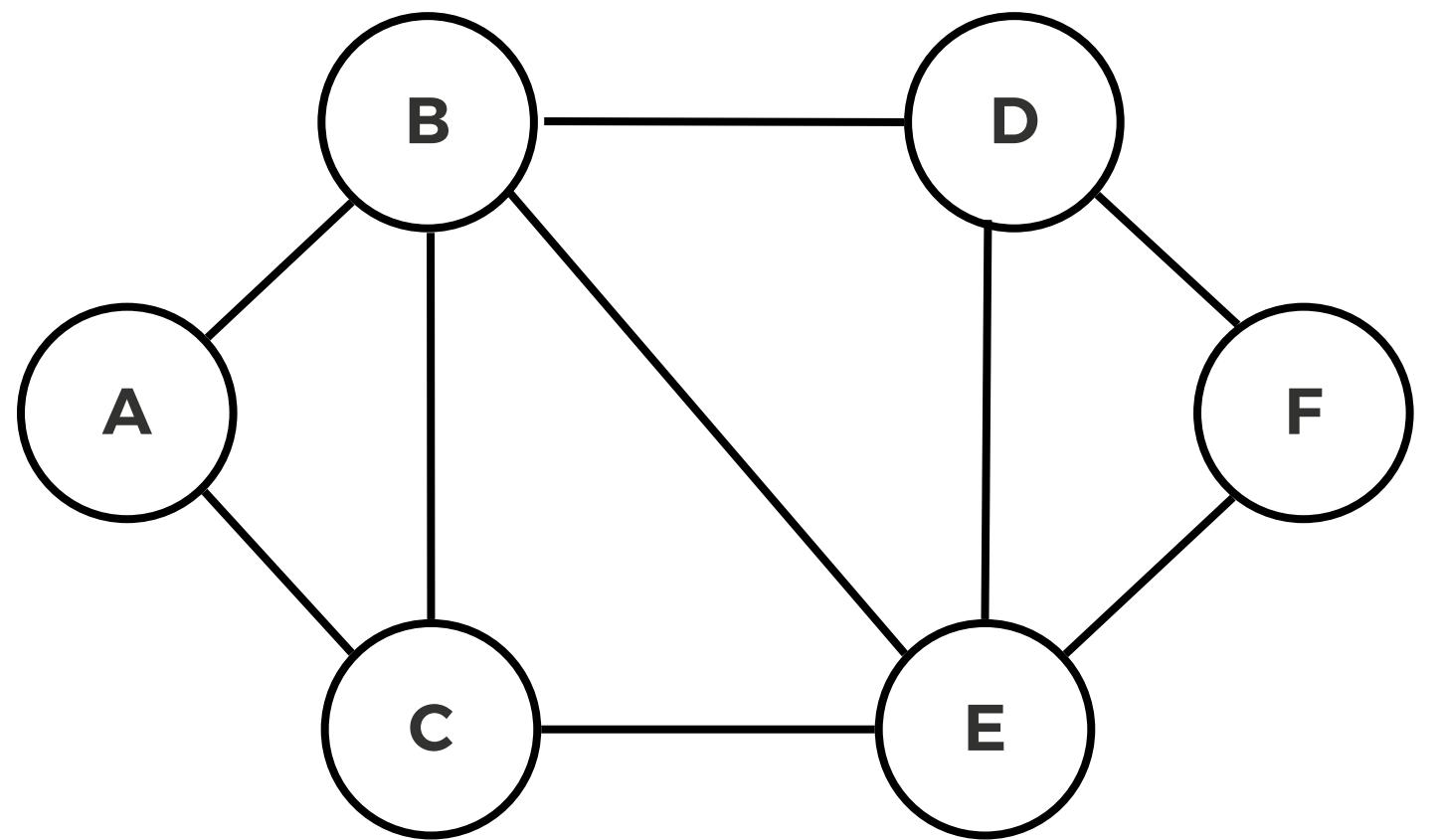
✿ PRINCIPE DE FONCTIONNEMENT.

💡 Exemple :

- En utilisant l'algorithme de DSATUR trouver la coloration des sommets
- 👉 $X=\{A,B,C,D,E,F\}$ est l'ensemble des sommets d'un graphe G.
- 👉 $E=\{\{A,B\},\{A,C\},\{B,D\},\{B,E\},\{B,C\},\{C,E\},\{D,E\},\{D,F\},\{E,F\}\}$ est l'ensemble des arêtes entre les sommets adjacents d'un graphe G.
- 👉 $DSAT(i)$ = nombre de couleurs différentes utilisées par les sommets adjacents au sommet i

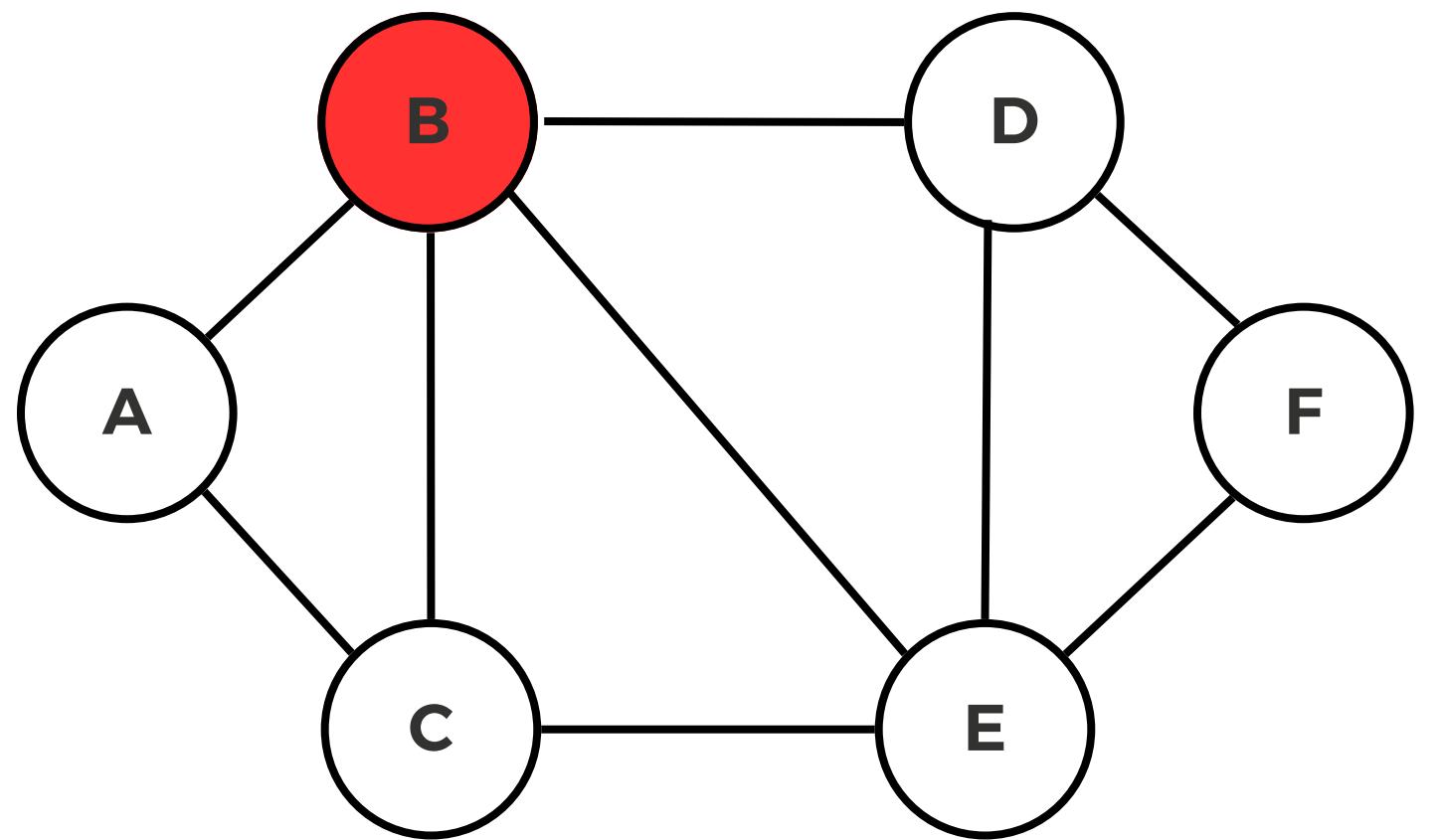


✿ PRINCIPE DE FONCTIONNEMENT.



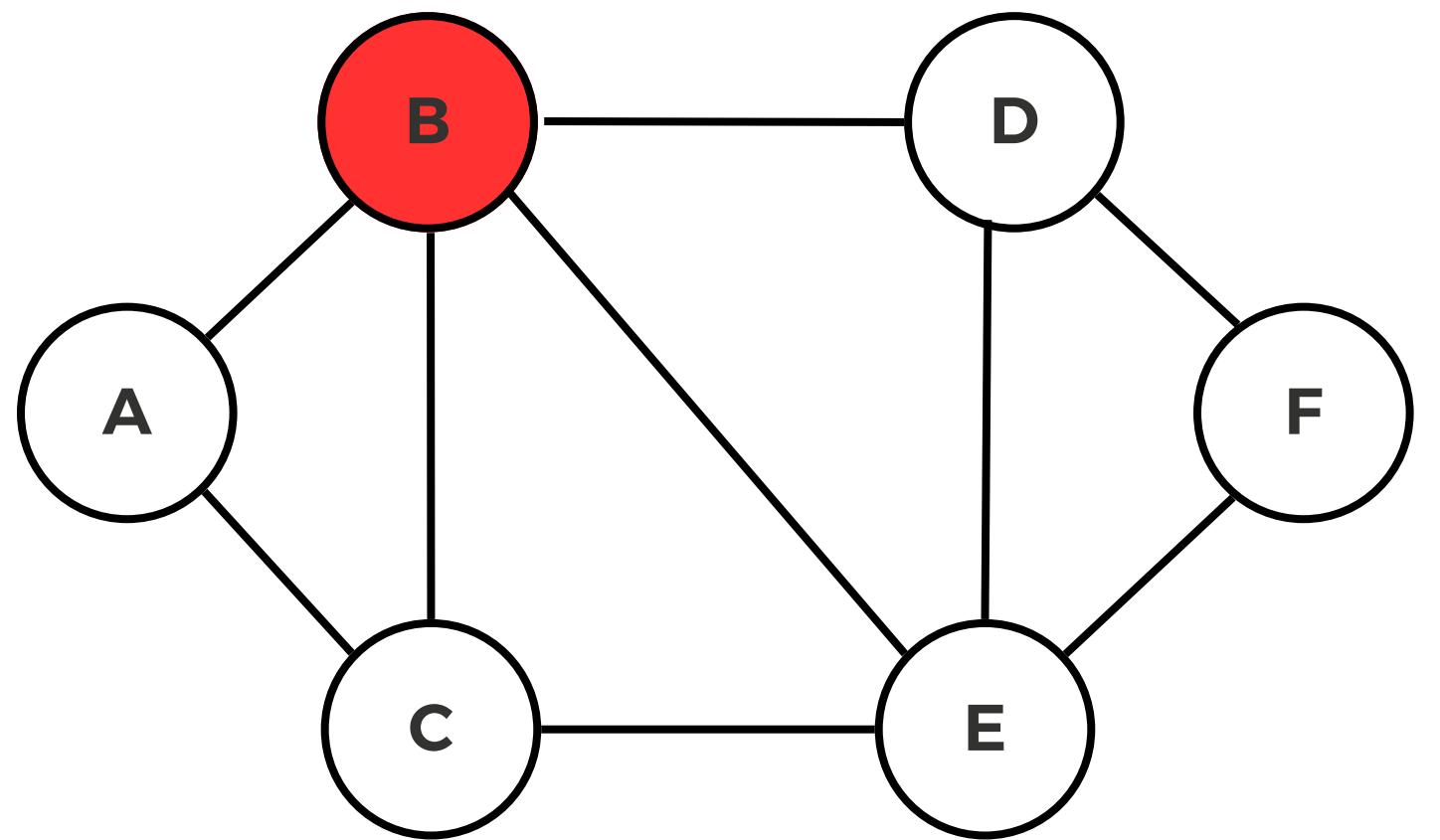
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2

✿ PRINCIPE DE FONCTIONNEMENT.



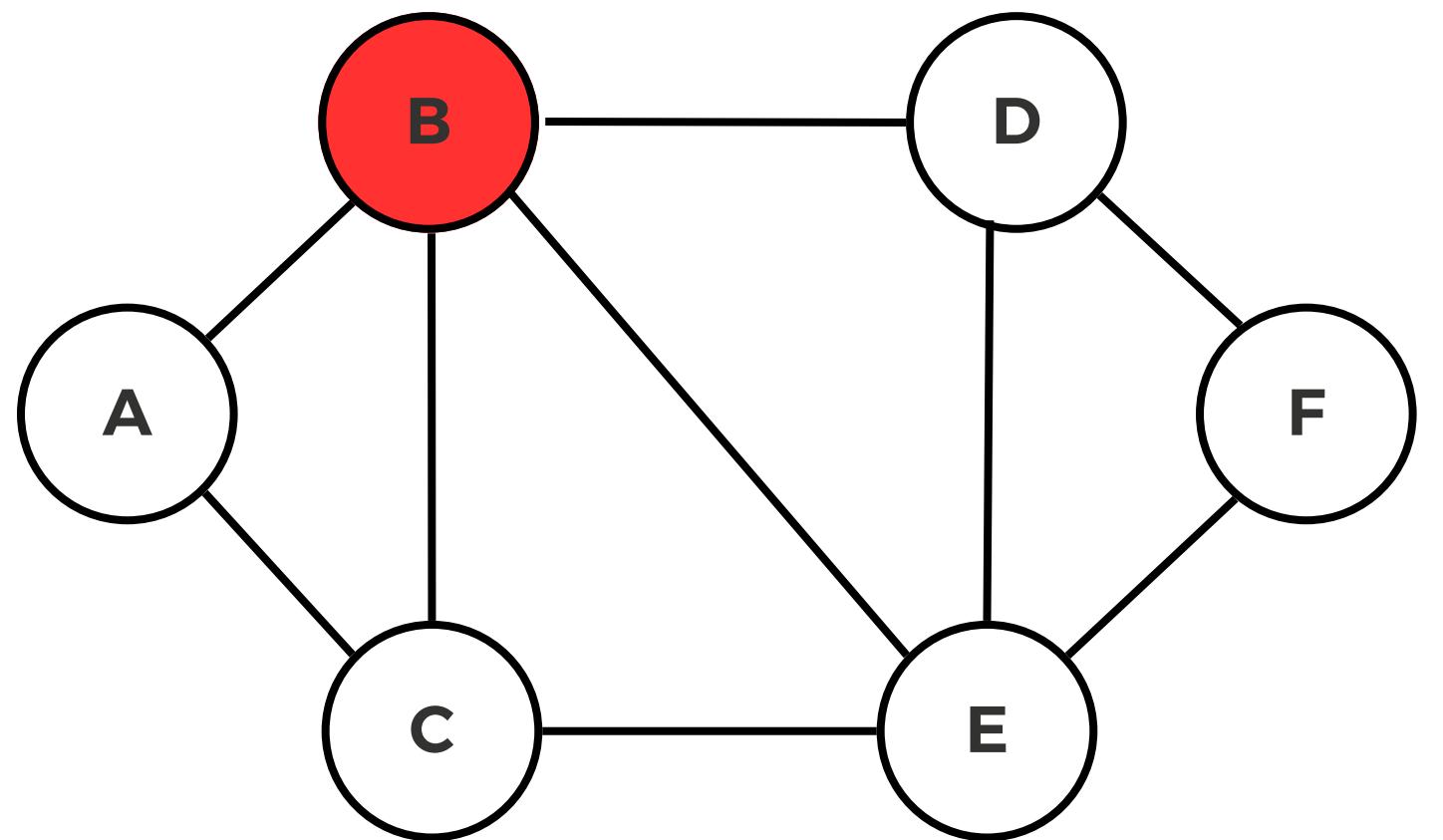
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R					

✿ PRINCIPE DE FONCTIONNEMENT.



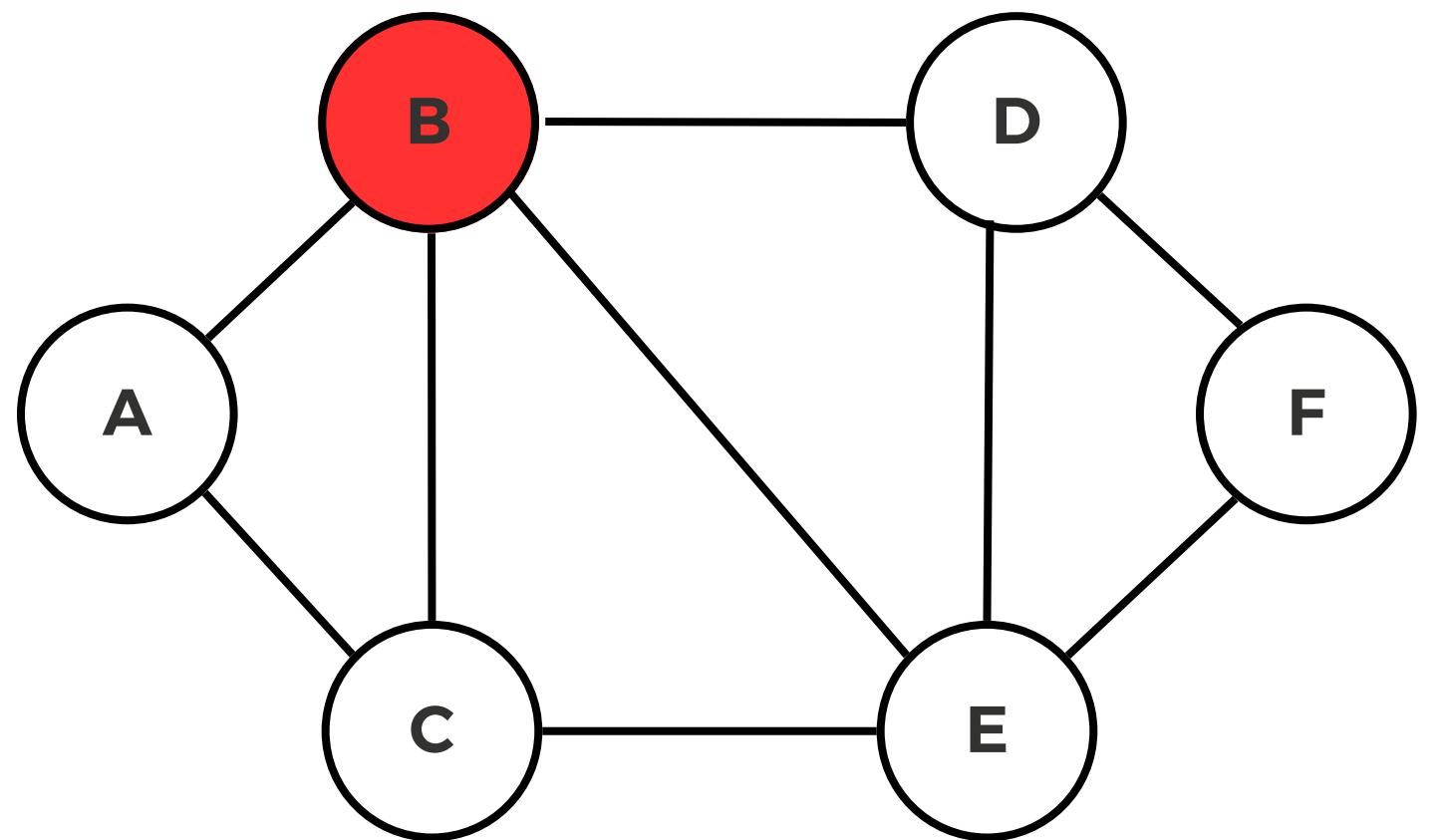
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1				

✿ PRINCIPE DE FONCTIONNEMENT.



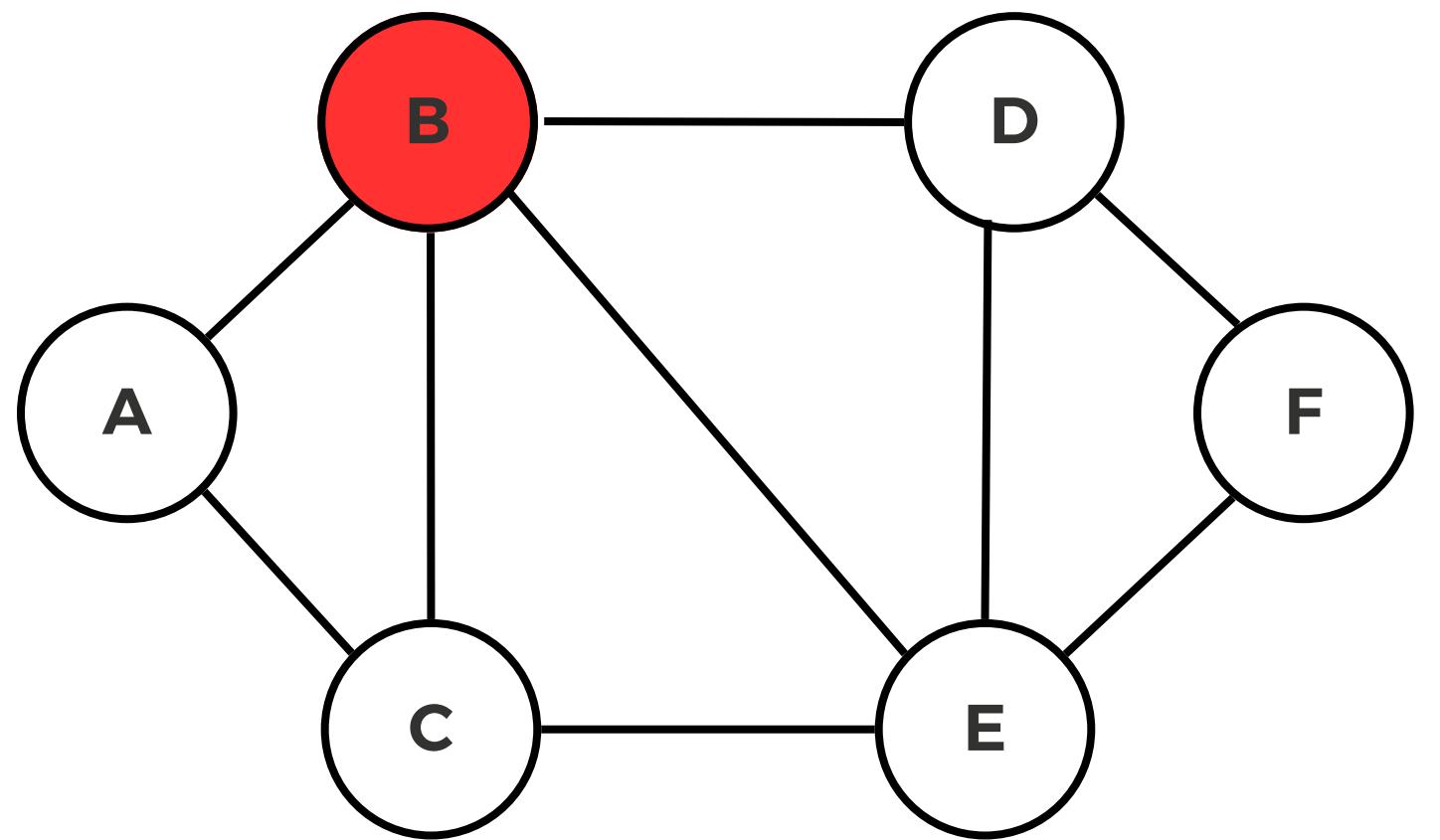
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
cl=R	1	1	1			

✿ PRINCIPE DE FONCTIONNEMENT.



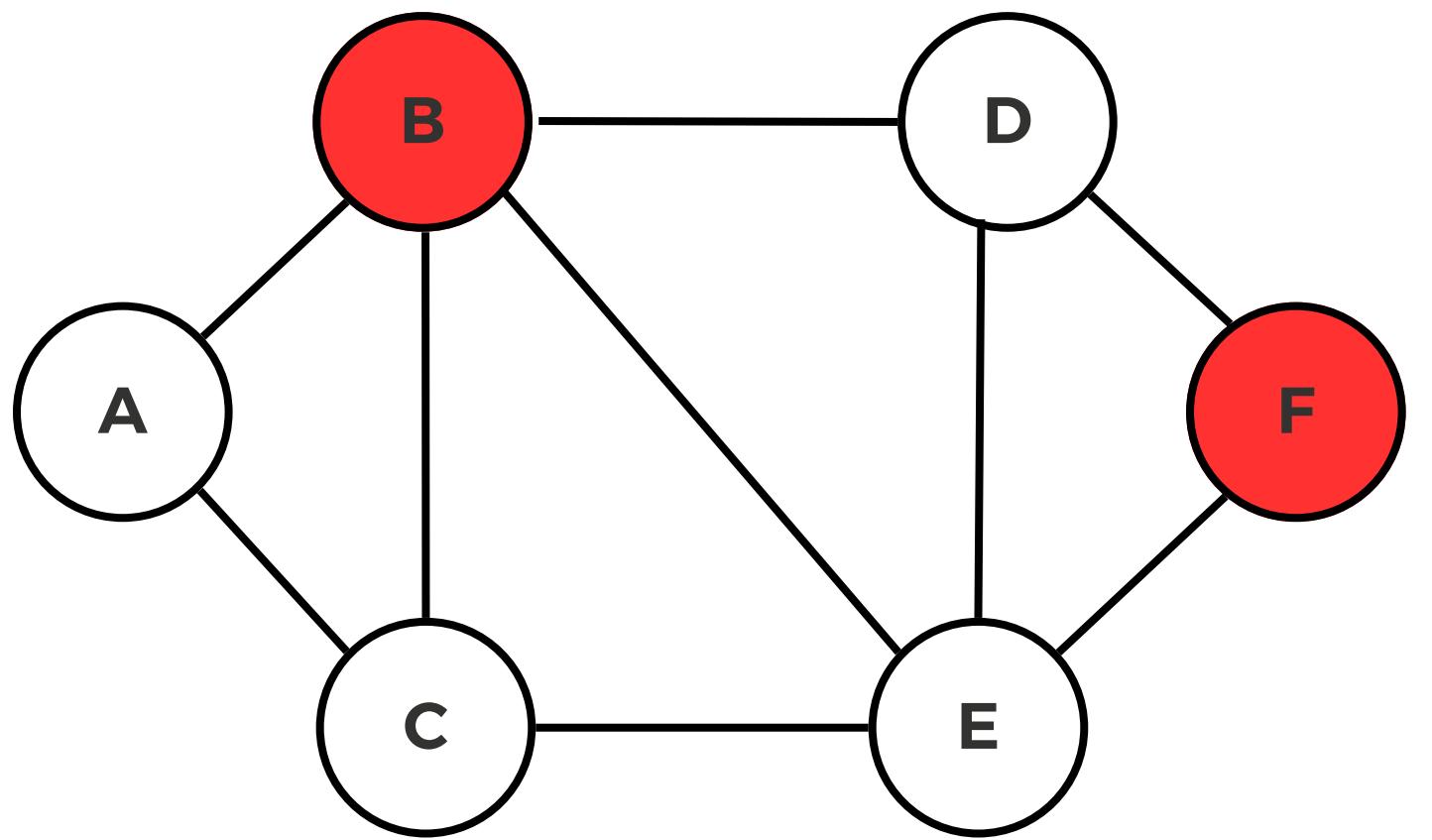
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
cl=R	1	1	1	1		

✿ PRINCIPE DE FONCTIONNEMENT.



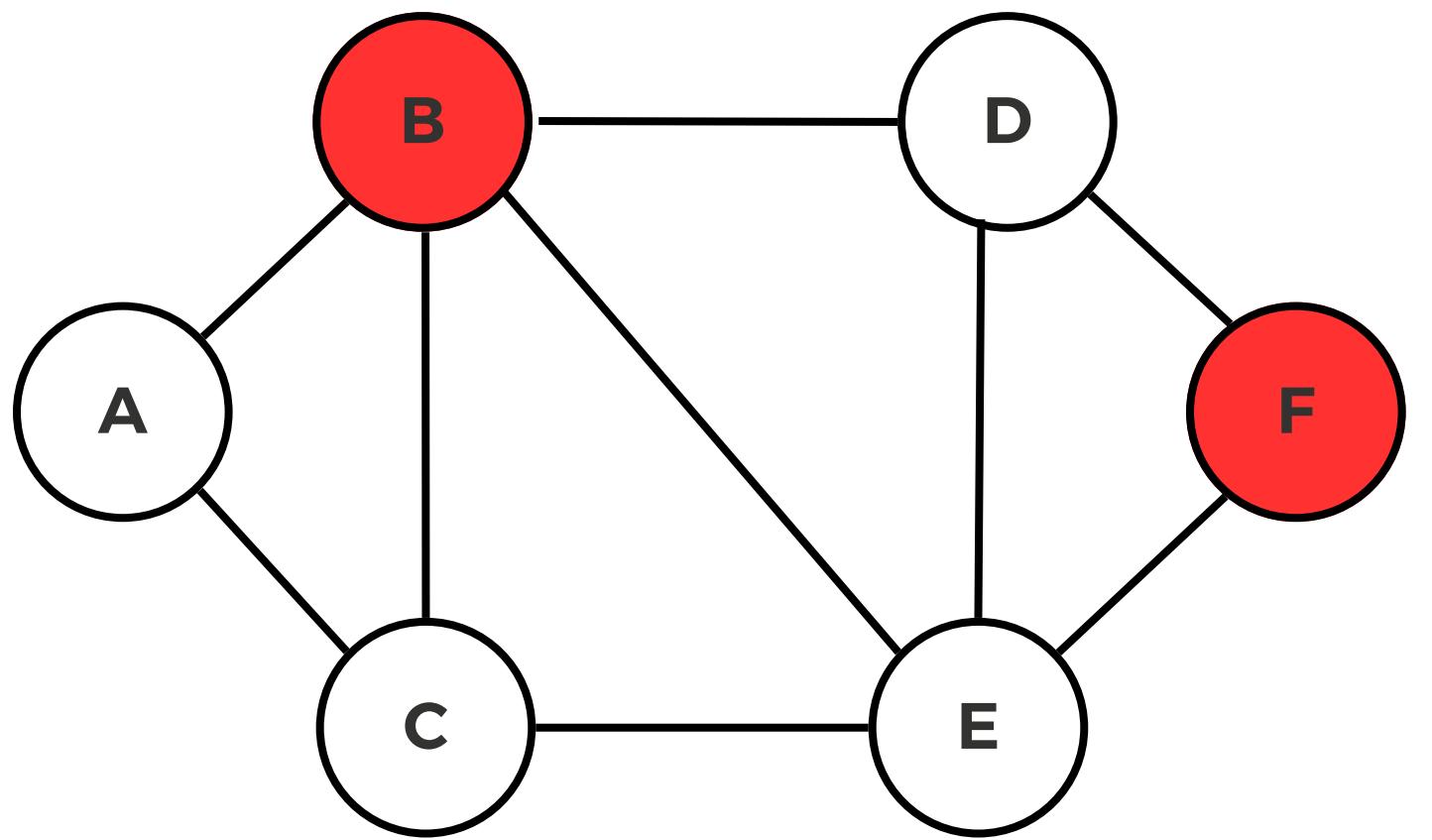
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
cl=R	1	1	1	1	1	

✿ PRINCIPE DE FONCTIONNEMENT.



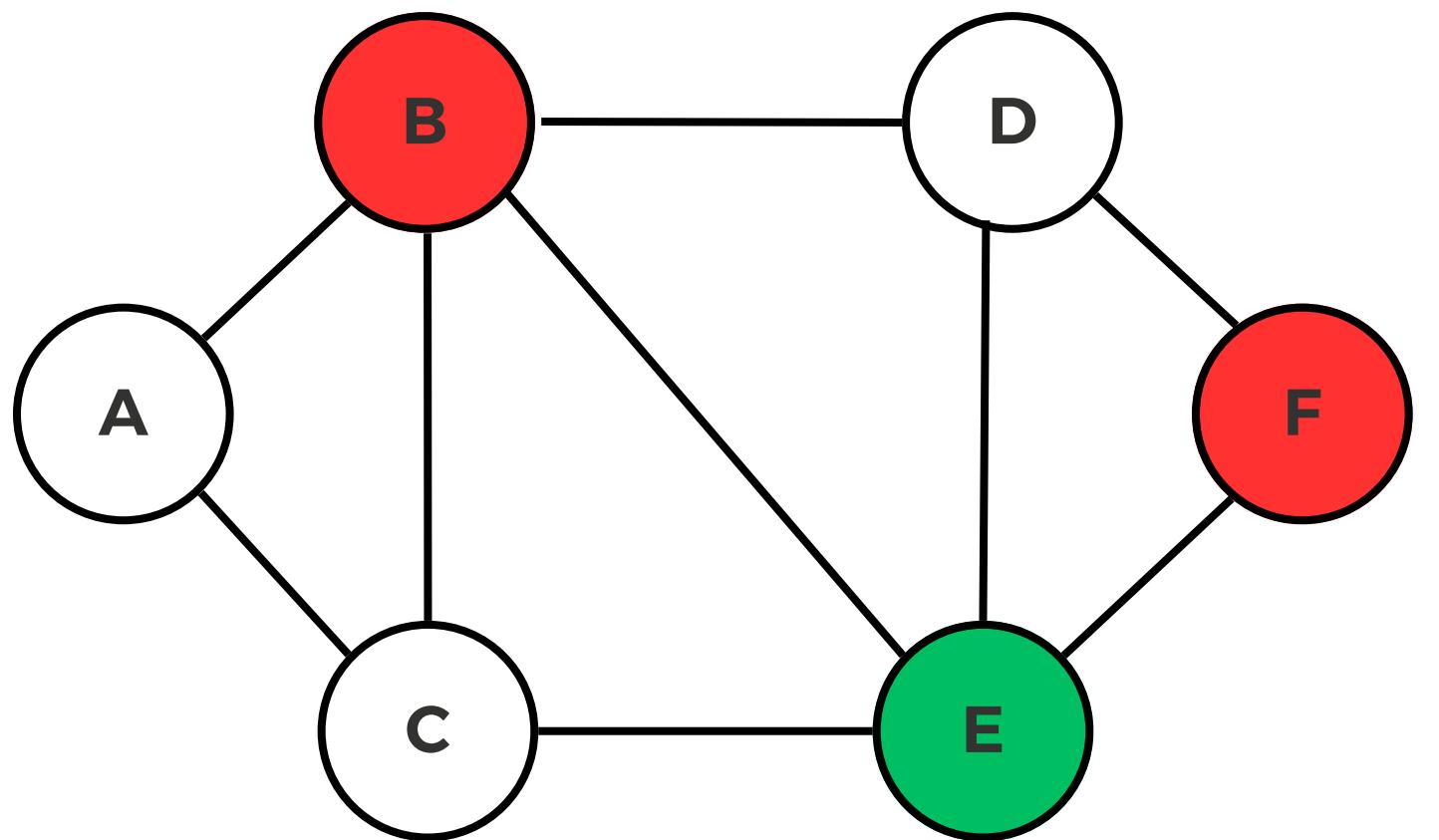
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R

✿ PRINCIPE DE FONCTIONNEMENT.



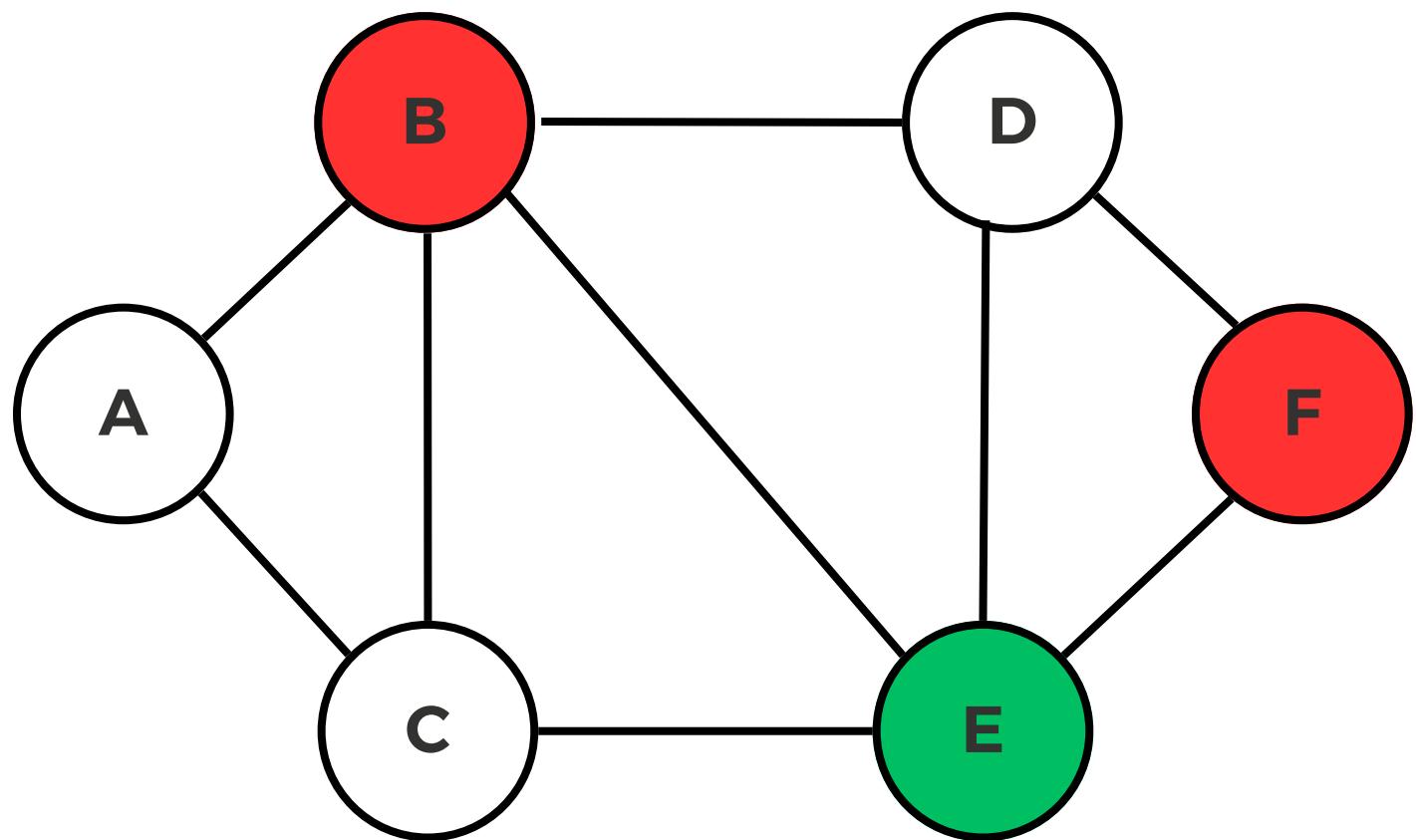
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R

✿ PRINCIPE DE FONCTIONNEMENT.



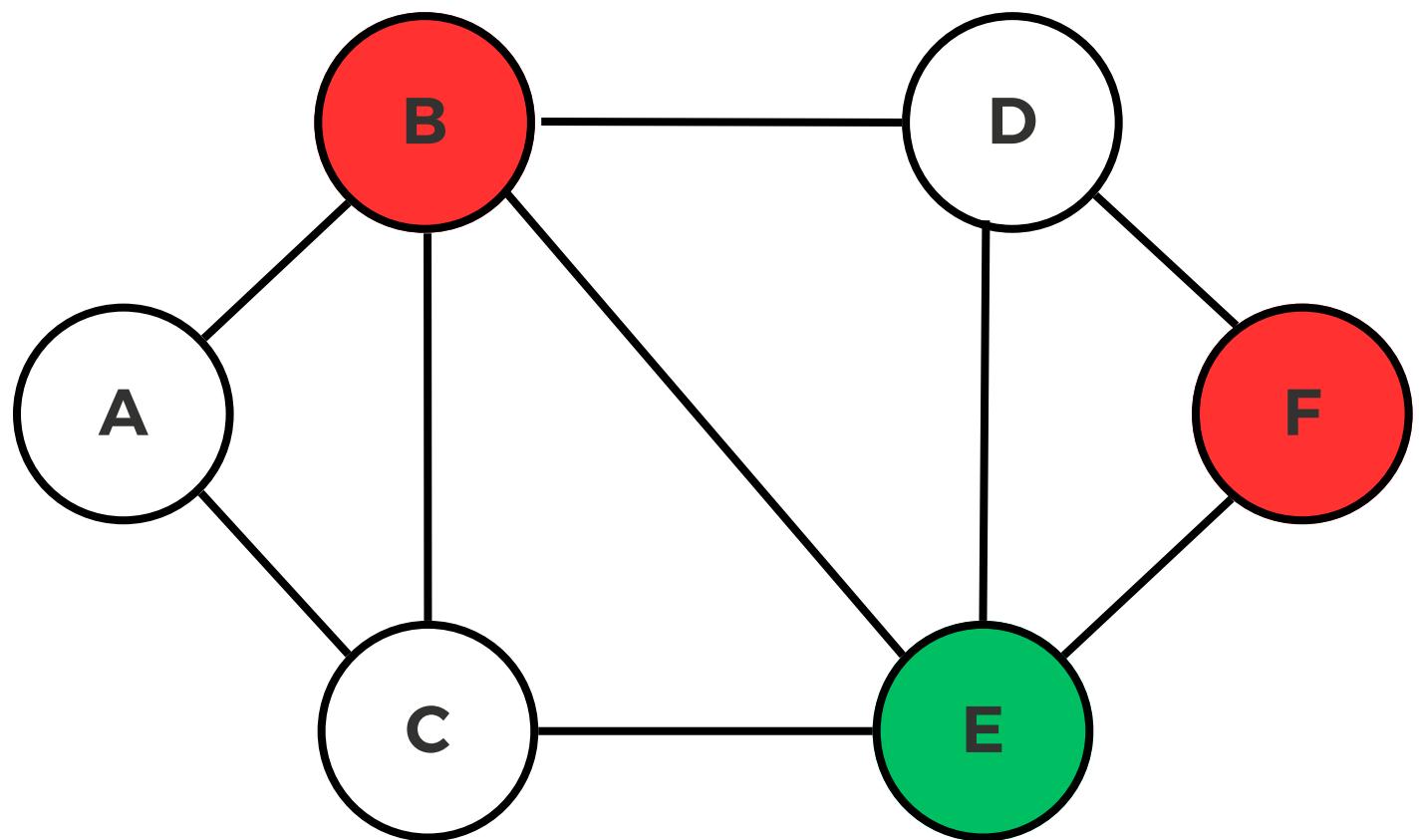
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R
3			c2=V			

✿ PRINCIPE DE FONCTIONNEMENT.



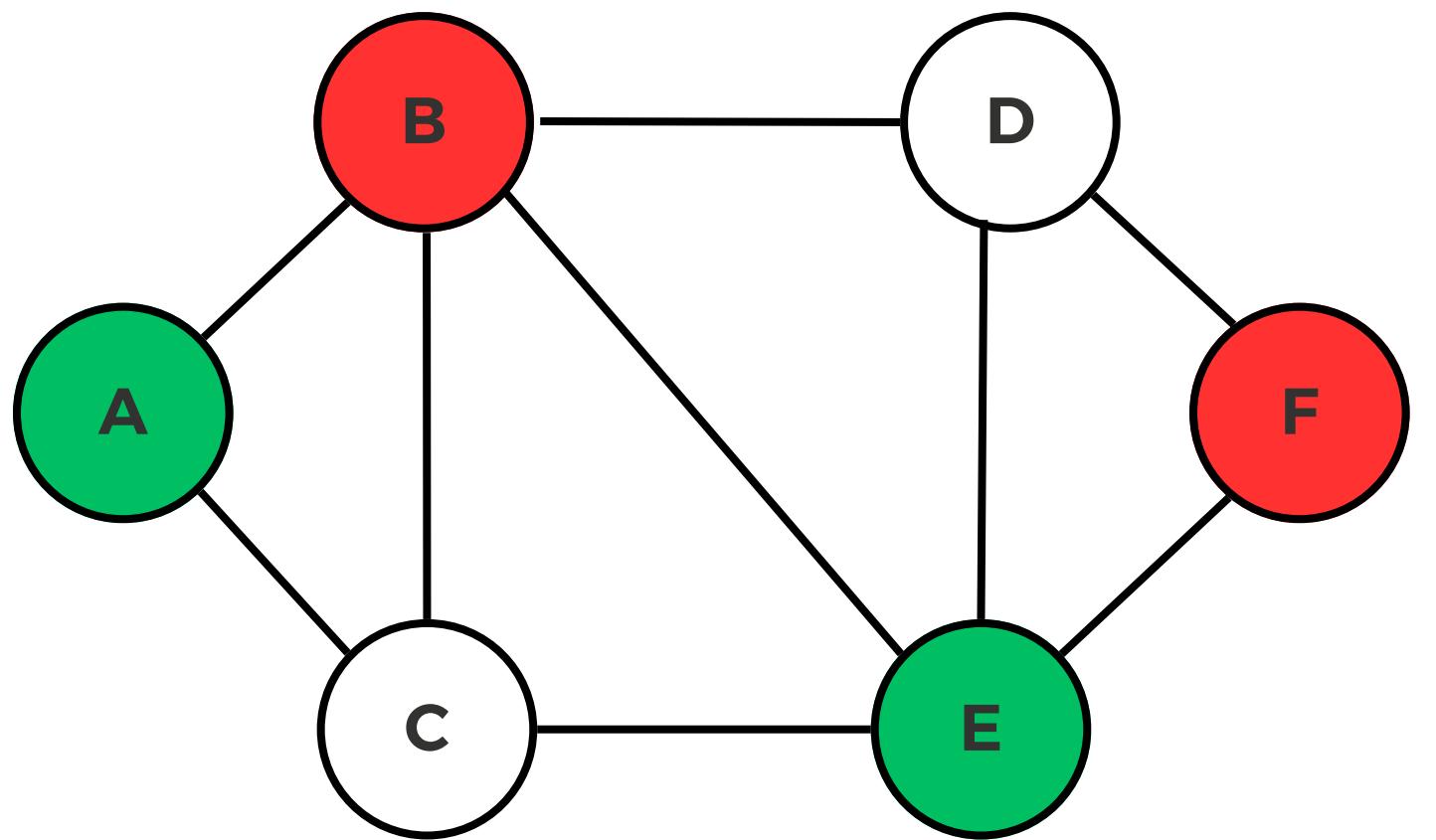
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R
3		c2=V		2		

✿ PRINCIPE DE FONCTIONNEMENT.



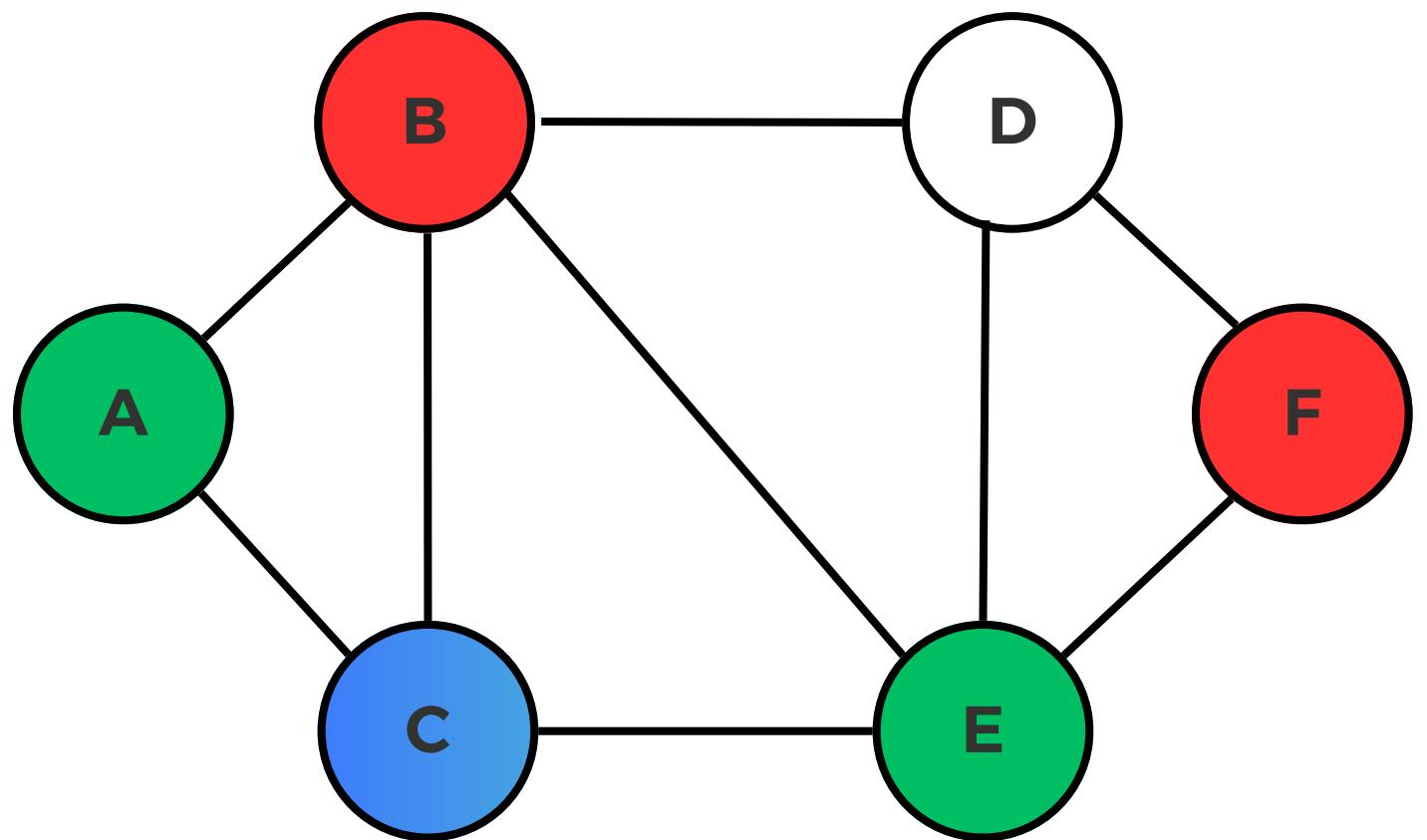
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R
3		c2=V		2	2	

✿ PRINCIPE DE FONCTIONNEMENT.



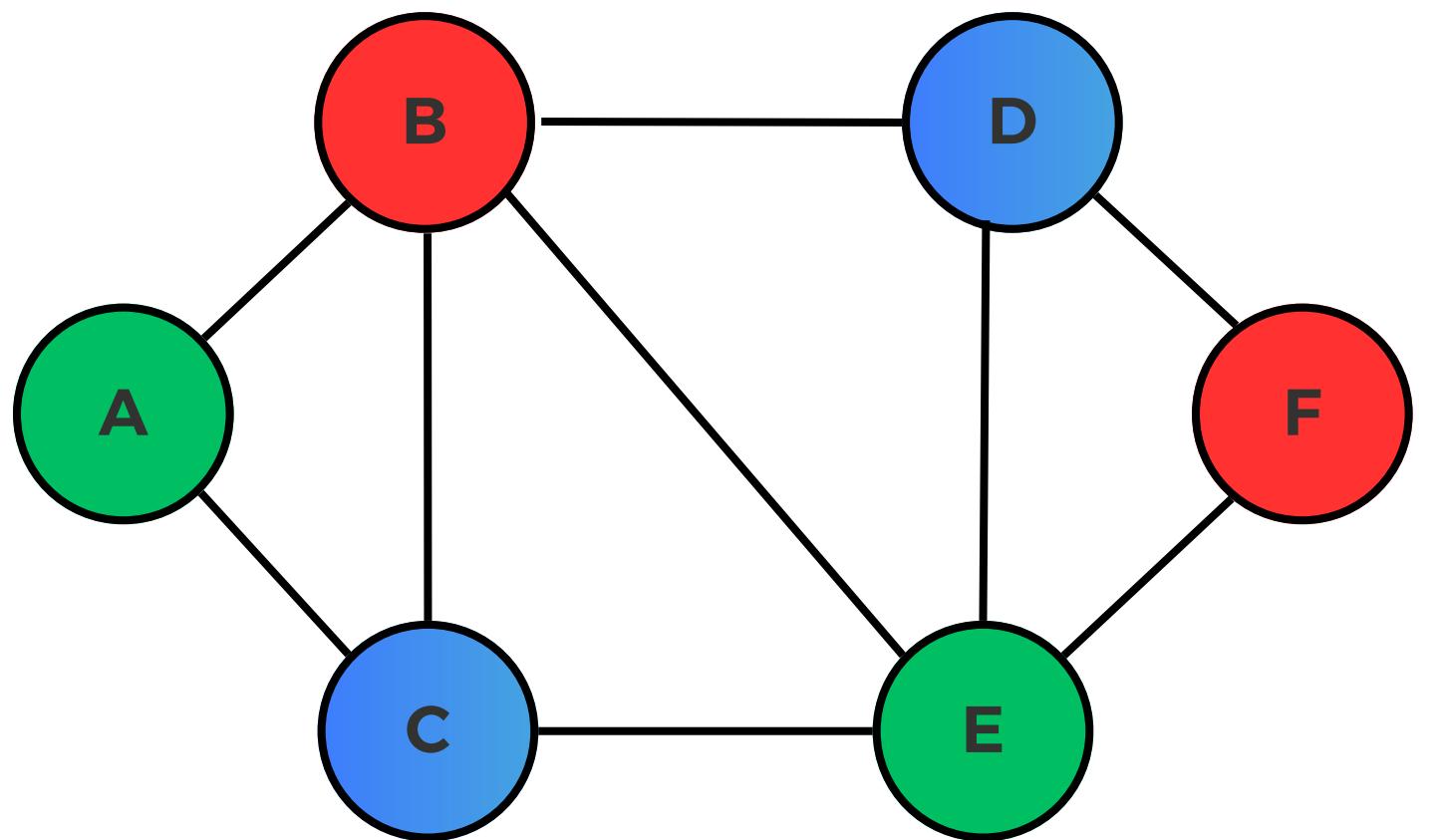
sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R
3		c2=V	2	2		
4						c2=V

✿ PRINCIPE DE FONCTIONNEMENT.



sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R
3			c2=V	2	2	
4						c2=V
5				c3=B		

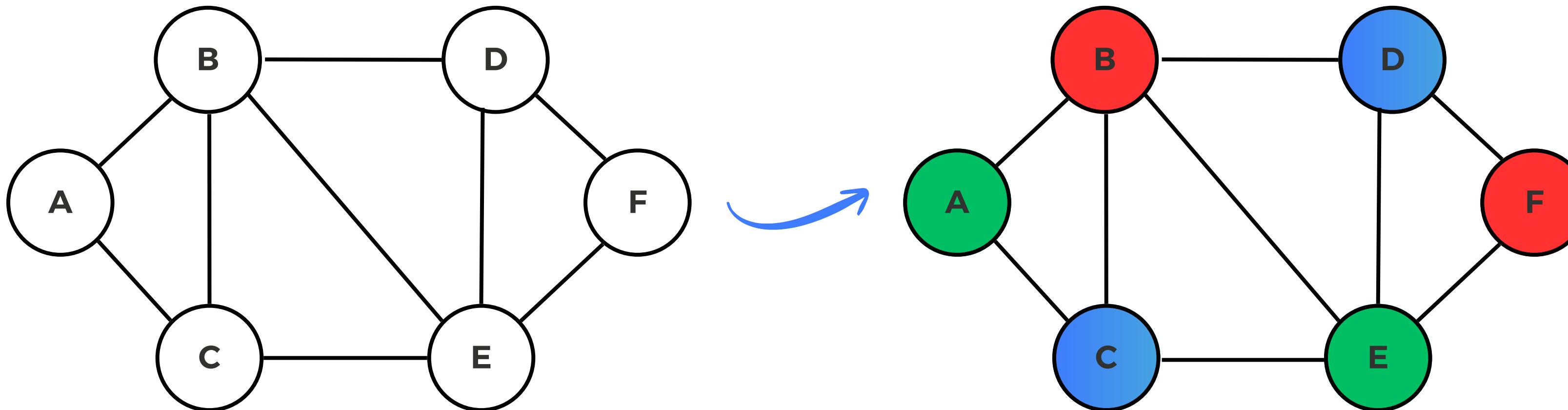
✿ PRINCIPE DE FONCTIONNEMENT.



sommet	B	E	C	D	A	F
Degré	4	4	3	3	2	2
1	c1=R	1	1	1	1	
2						c1=R
3			c2=V	2	2	
4						c2=V
5				c3=B		
6					c3=B	

✿ PRINCIPE DE FONCTIONNEMENT.

👉 Graphe coloré trouvé:



3-colorable
Graphe d'ordre 4

❖ IMPLEMENTATION

Entrée - Un graphe quelconque $G = (V, E)$

Sortie - Une coloration propre $c : V \rightarrow \mathbb{N}$

1. $n := |V|;$
2. Couleur := Tableau (Taille : n) (Défaut : -1);
3. Deg := Tableau (Taille : n) (Défaut : 0);
4. Dsat := Tableau (Taille : n) (Défaut : 0);
5. # Initialisation #
6. **Pour** i de 0 à $n - 1$ **faire**
 7. Deg.(i) $\leftarrow d(i);$
 8. Dsat.(i) $\leftarrow d(i)$
9. **Fin Pour**;
10. # Boucle principale #
11. **Pour** i de 0 à $n - 1$ **faire**
 12. $x := 0;$
 13. # Recherche du sommet de Dsat max #
 14. **Pour** y de 0 à $n - 1$ **faire**
 15. **Si** Couleur.(x) = -1 & Dsat.(y) > Dsat.(x) **faire**
 16. $x := y$

❖ IMPLEMENTATION

```
17.          Fin Si
18.          Fin Pour;
19.          # En cas de conflit, prendre celui de Deg max #
20.          Pour  $y$  de 0 à  $n - 1$  faire
21.              Si Couleur.( $x$ ) = -1 & Dsat.( $y$ ) = Dsat.( $x$ ) & Deg.( $y$ ) > Deg.( $x$ )
22.                   $x := y$ 
23.          Fin Si
24.          Fin Pour;
25.          # Recherche de la plus petite couleur non utilisée dans  $\mathcal{N}(x)$  #
26.          Libre := Tableau (Taille :???) (Défaut : vrai);
27.          Pour  $y$  dans  $\mathcal{N}(x)$  faire
28.              Si Couleur.( $y$ ) ≠ -1 faire
29.                  Libre.(Couleur.( $y$ )) ← faux;
30.              Fin Si
31.          Fin Pour;
32.          index := 0;
33.          Tant Que Libre.(index) = faux faire
34.              index := index + 1
35.          Fin Tant Que;
```

❖ IMPLEMENTATION

```
36.          # Mettre à jour Dsat pour les voisins de x #
37.          Pour  $y$  dans  $\mathcal{N}(x)$  faire
38.              Si  $Dsat.(y) = Deg.(y)$  faire
39.                   $Dsat.(y) \leftarrow 1;$ 
40.                  Si  $index \notin couleur(\mathcal{N}(y))$  faire
41.                       $Dsat.(y) \leftarrow Dsat.(y) + 1;$ 
42.                  Fin Si
43.              Fin Pour;
44.          # Affecter la couleur donnée par index à x #
45.          Couleur.(x)  $\leftarrow index$ 
46.      Fin Pour;
47.      Renvoyer Couleur
```

✿ COMPLEXITÉ.

★ Dsatur:

La formule:

$$T_{DSAT} = T_{deg} + T_{int\ DSAT} + T_{trier\ DSAT} + n * (T_R\ calculer + T_{MAJ})$$

Alors:

$$T_{DSAT} = O(1) + O(1) + O(1) + n * (O(n) + O(n))$$

$$T_{DSAT} = O(1) + O(n^2)$$

Donc:

$$T_{DSAT} = O(n^2)$$

MERCI POUR VOTRE ATTENTION