I. Rappels et définitions

a. Quelques notions de graphe

L'algorithme de Dijkstra est un algorithme permettant de déterminer les plus courts chemins entre certains points d'un graphe. Avant de détailler fonctionnement de cet algorithme, commençons par introduire le vocabulaire sur les graphes.

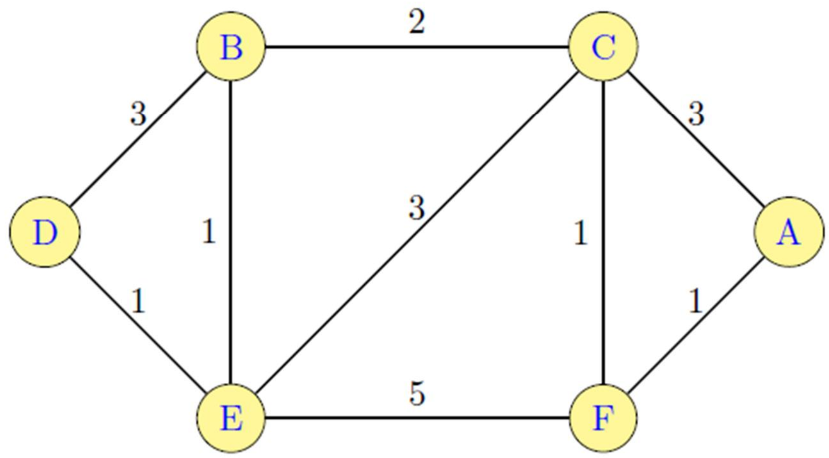
b. Définition

On appelle graphe G= (S, A) un couple où :

- S est un ensemble appelé ensemble des sommets des arrêts.
- A est un ensemble, appelé ensemble des arrêts.

Dans la suite, on supposera S fini

Exemple : On considère le graphe suivant, donné par sa représentation graphique.



Le graphe $\mathcal{G} = (S, A)$ est ici défini par :

× l'ensemble des sommets $S = \{A, B, C, D, E, F\}$,

× l'ensemble des arêtes $\mathcal{A} = \{(A, B), \ldots, (E, F), \ldots, (F, A)\}$.

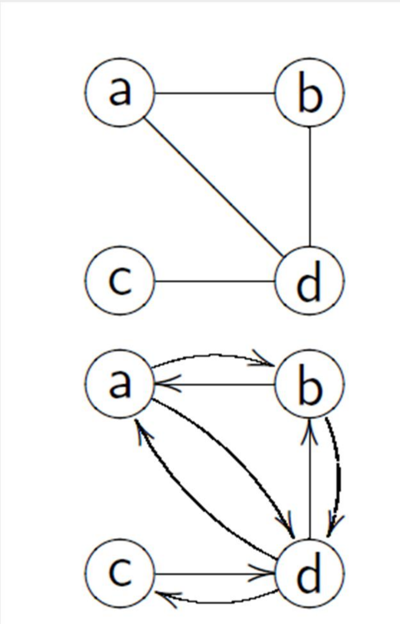
Remarque

Le graphe présenté ici possède les caractéristiques suivantes.

- Il n'est pas **orienté** : si $(u, v) \in \mathcal{A}$ alors on a $(v, u) \in \mathcal{A}$. Ainsi, si on peut aller de u à v, alors on peut aussi aller de v à u.
- Il est **simple** : il y a au plus une arête entre deux sommets.
- Il est **pondéré** : à chaque arête on associe une valeur positive appelée poids. Ce poids représente la distance entre les deux sommets.

II. Graphes et chemins

Un **graphe non orienté** est un couple < S, A>, où S est un ensemble fini non vide et A un ensemble de couple non ordonnés d'élément s de S. Un élément de A est appelé une **arête.** Une arête est représentée par un ensemble de deux sommets. Un sommet qui n'apparait dans aucune arête est dit **isolé**. Si $\{a\ ,b\} \epsilon \mathrm{A}$, les sommets a et b sont dits **adjacents**.



G= <S, A > ou S= $\{a, b, c\ d\}$ où

S=$\{a, b, c, d\}$
A=$\{\{a, b\}, \{a, d\}, \{b, d\}, \{c, d\}\}$

G'= <S, A' > où

S=$\{a, b, c, d\}$
A'=$\{\{a, b\}, \{b, a\}, \{a, d\}, \{d, a\}, \{b, d\}, \{d, b\}, \{c, d\}, \{d, c\}\}$

Remarque : un graphe non orienté G est une représentation équivalente par un graphe orienté G'

Chemins

Un chemin dans un graphe orienté est une séquence (suite) de sommet et d'arcs telle que :

i) La séquence débute par un sommet et se termine par un sommet ;
ii) Les sommets et les arcs alternent ;
iii) Chaque arc est précédé par son sommet origine et est suivi par son sommet cible ;
iv) Aucun arc n'apparait plus d'une fois.

Exemple 1 : dans la figure ci- contre

$$\langle c, \langle c, e\rangle, e, \langle e, d\rangle, d\rangle$$

et

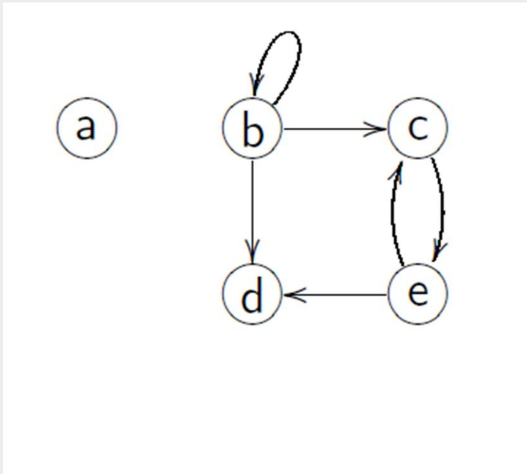
$$\langle b, \langle b, c\rangle, c, \langle c, e\rangle, e\langle e, c\rangle, c\rangle$$

sont des chemins, mais pas

$$\langle b, \langle b, c\rangle, c, \langle c, e\rangle, e\langle e, c\rangle, c, \langle c, e\rangle, e\rangle$$

ni

$$\langle b, \langle b, c\rangle, c, \langle e, c\rangle, c\rangle \;.$$

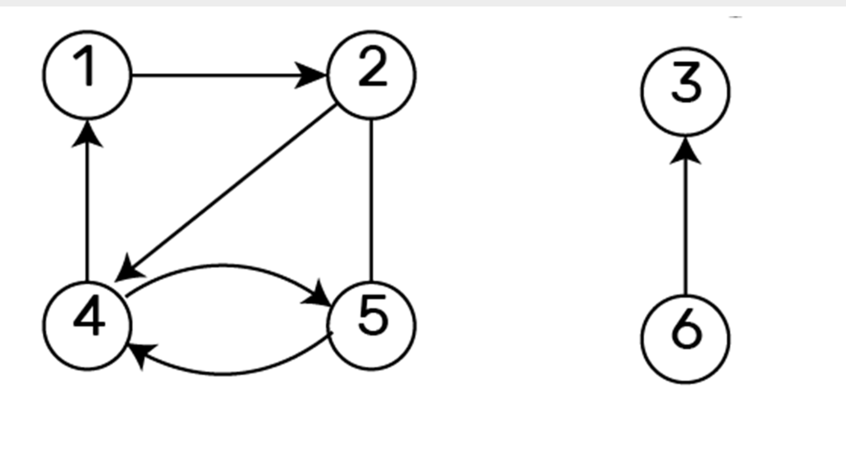


La longueur d'un chemin est le nombre d'arcs de ce chemin :

Longueur de <c, < c, e >, e >=1 et longueur de <c> = 0

- Un chemin est un **Cycle** s'il contient au moins un arc et que le premier et le dernier sommet du chemin sont identiques.
- Un chemin est **simple** si tous ses arcs sont distants. Un chemin est **élémentaire** si ses sommet (sauf éventuellement le premier et le dernier) sont tous distants.
- Dans les graphes non orientés, on définit la notion correspondante de **chaine**

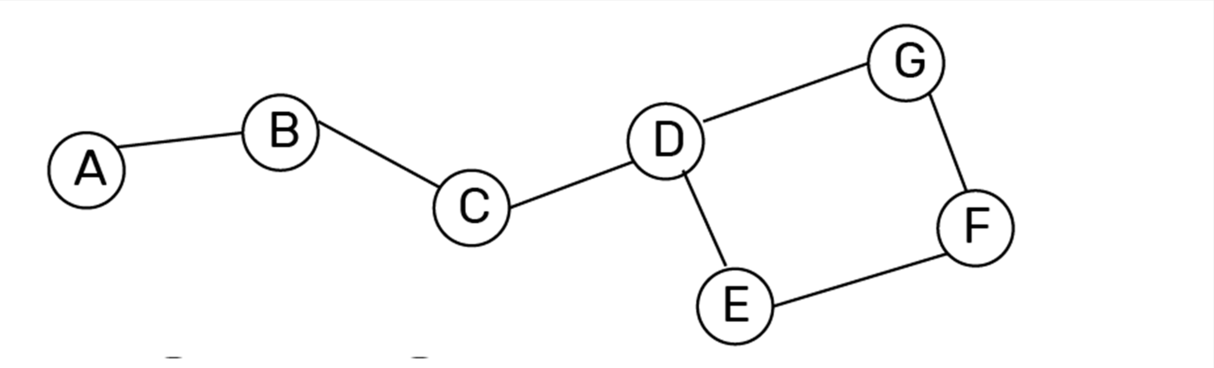
Exemple 2 : Dans la figure suivante, le chemin (1 ;2 ;5 ;4) est élémentaires et de longueur 3, mais le chemin (2 ;5 ;4 ;5) n'est pas élémentaire.



Remarque : Dans tout ce qui suit, nous ne considérons uniquement que les chemins simples.

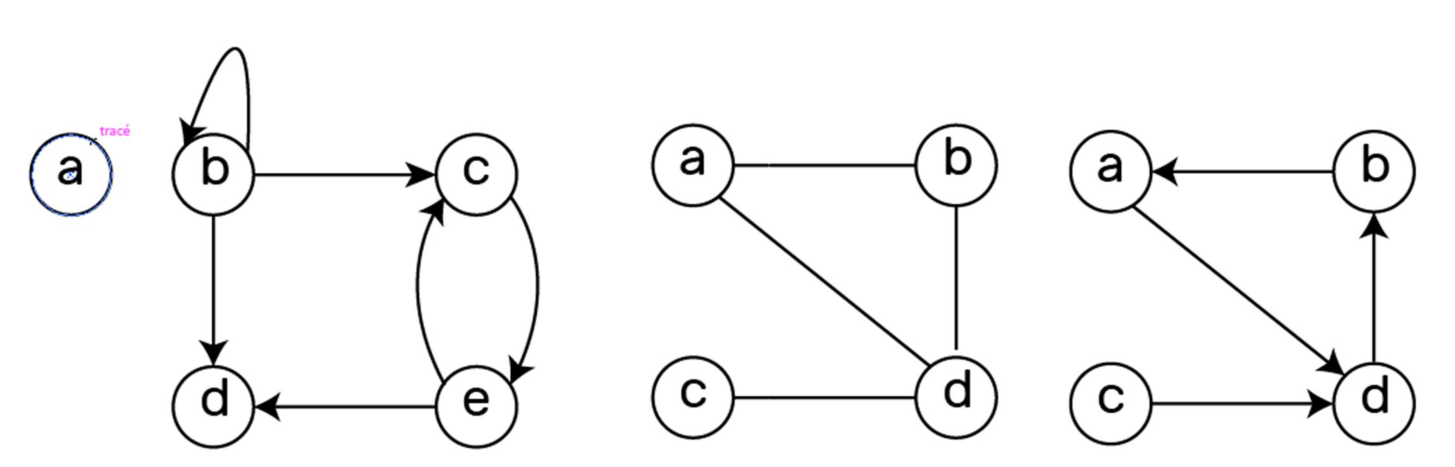
- **La longueur** d'un chemin est le nombre d'arcs (ou arrêts) qui composent ce chemin.
- **La distance** entre deux nœuds est le minimum des longueurs parmi tous les chemins qui relent ces deux nœuds.

Exemple : la suite d'arrêtes {B, C} ; {C, D} ; {D, E} ; {E, F} ; {F, G} du graphe non orienté ci-dessous forme un chemin de longueur. La distance entre les nœuds B et G est toute fois 3, car la longueur du chemin.



{B, C} ; {C, D} ; {D, G} est plus courte

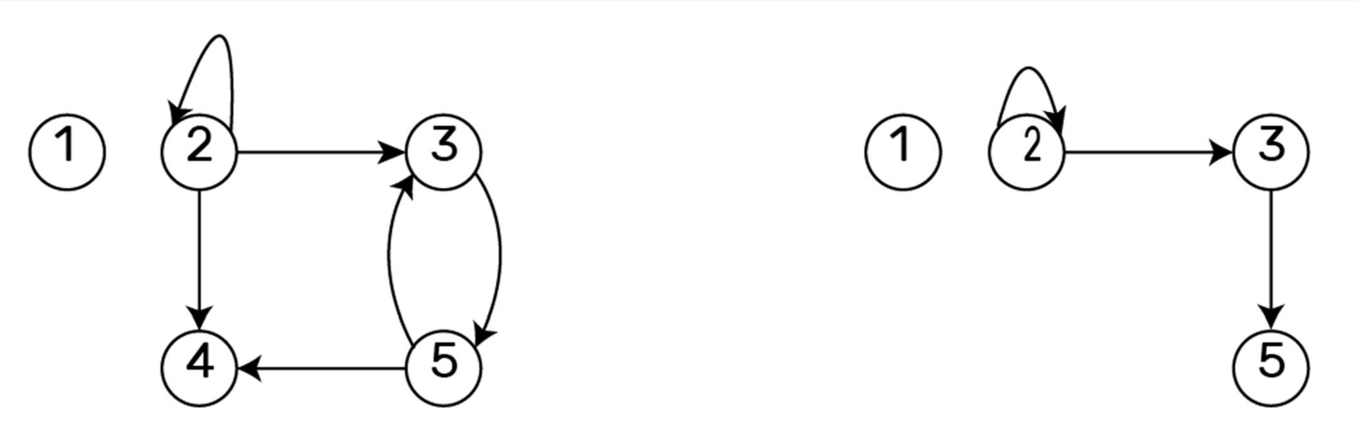
- Un graphe non orienté est dit **connexe** s'il y'a un chemin entre n'importe quelle paire de sommets. Un graphe orienté est dit **connexe** si, en transformant son arc en arrêts non orientées, on obtient un graphe orienté connexe.
- Par exemple, le graphe de gauche ci-dessous n'est pas connexe, mais les deux autres le Sont.



1. Sous-graphes

Un graphe <S', A'> est un **sous graphe** du graphe <S, A> ssi $S' \subseteq S$ et A'$\subseteq A$.

Notez que parce que <S', A'> est un graphe, les extrémités des arcs ou des arrêts de A' sont dans S'. Le graphe de droite ci-dessous est un sous graphe du graphe de gauche.



NB : Cette notion s'applique aussi aux graphes non orientés

2. Graphes pondérés

a. Définition

On nomme graphe pondéré tout graphe étiqueté tels que toutes les étiquètes sont des nombres positifs.

Exemple : pour un graphe représentant une carte routière on peut considère pour étiquettes les distances en km ou le temps en heures ou minuties ou le prise des péages en francs.

b. Remarques

- le nombre positifs affecté à une arrête(ou un are).se nomme poids de cet arrête (ou de cet arc).
- le poids d'une chaine(ou d'un chemin) est la somme des poids des arrêtes (ou des ars)qui la composent.

Une plus courte chaine (ou plus cours chemin)

Entre deux sommets est parmi les chaines (ou les chemins) qui relient ces deux sommets une chaine .de poids minimal.

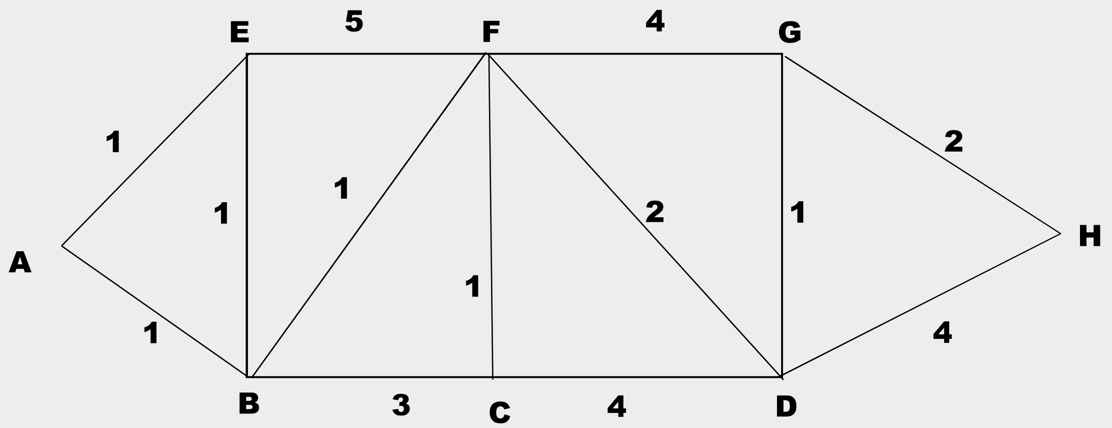
3. Algorithme de Dijkstra

Mathématicien hollandais (1930-2002) pernet de déterminer le trajet de points minimal ou la plus courte chaine (chemin) entre 2 sommet fixes .

3.1 Principe

- Si $s_1\ s_2\ s_3 \ldots s_p$ (avec p un entier naturel non nul) est le plus court chemin reliant s_1 à s_p alors pour tous entier **i** compris entre **1 et p,** $s_1\ s_2 \ldots s_i$ est un court chemin reliant s_1 à s_i donc on déterminer de proche en proche les chemins minimaux reliant s1aux autres sommets **1** jusque s_p.
- Chaque interaction de **l'algorithme de Dijkstra** Pernet de remplir une ligne d'un tableau dont la demi ligne donnera un chemin minimal.

Exemple : Soit le graphe pondéré non orienté et étiqueté d'ordre 8 suivant



3.2 Algorithme

Étape 1 (initialisation)

- Dans la première ligne du tableau on écrit tous les sommets du graphe. (L'ordre des sommets est arbitraire. Ici pour l'exemple on place en premier le sommet initial et en dernier le sommet final entre eux les sommets sont placés par ordre alphabétique).
- Dans la deuxième ligne on écrit sous le point initial : 0, sous les autres sommets on écrit : ∞ (correspondant au poids affecté aux sommets).

Étape 2

- Choisir parmi les sommets (non encore marqués) un sommet X de poids minimal (si plusieurs sommets ont le même poids minimal alors le choix parmi ces sommets est arbitraire).
- Dans la nouvelle ligne et dans la colonne X, on marque définitivement ce poids minimal et après cette case de la colonne on n'écrira plus rien (pour l'exemple on mettra en couleur les cases suivantes).

Étape 3

Pour tous les sommets Y adjacents à X qui ne sont pas définitivement marqués, on calcule la somme σ du poids du poids de X et du poids de l'arête (ou de l'arc) reliant X à Y.

- Si cette somme σ est strictement inférieure au poids de Y alors on écrit dans la case de la ligne et de la colonne Y comme poids la somme obtenue σ et on notera σ (X).
- Sinon, on écrit dans cette case le poids précédent.

Étape 4

- S'il reste des sommets non marqués définitivement alors **on repart à l'étape 2.**
- Sinon on passe à l'étape 5.

étape 5

On obtient le poids du plus court chemin dans la dernière ligne. Puis on détermine le plus court chemin obtenu.

Retour à l'exemple :

Initialisation

A	B	C	D	E	F	G	H	Sommet fixé
0	∞	∞	∞	∞	∞	∞	∞	A
X								
X								
X								
X								
X								
X								

Les X signifie que ces cases ne seront plus utilisées.

On part à la deuxième ligne en partant de A. On revient dans notre graphe, on remarque A est lié à B avec un poids de 3 et lié aussi à E avec un poids de 1. Par conséquent je vais utiliser le point B et le point E, le reste je mets infini (∞) i.e

Par suite, A $\longrightarrow$ E de poids 1 + 0 = 1

ie 1 (A)

En résumé

A $\longrightarrow$ B : $\sigma = 0 + 3 = 3$

A $\longrightarrow$: $\sigma = 0 + 1 = 1\ (A)$

Entre 3 et 1 le min $(3 , 1)$ = 1 le sommet fixe est donc E

Nous avons :

A	B	C	D	E	F	G	H	Sommet fixé
0	∞	∞	∞	∞	∞	∞	∞	A
X	3 (A)	∞	∞	1 (A)	∞	∞	∞	E
X				X				

Maintenant remplissons la ligne suivante en partant de E. On revient sur son graphe et regarde E. Alors, le point E est relié au point F et B et le reste se met à l'infini (∞) i.e

A	B	C	D	E	F	G	H	Sommet fixé
X	3 (A)	∞	∞	X	∞	∞	∞	E
X	2 (E)	∞	∞	X	6 (E)	∞	∞	

N B : 1 (A) le point minimal

E $\xrightarrow{1}$ B : $\sigma = 1 + 1 = 2$ = 2(E)

E $\xrightarrow{5}$ F : $\sigma = 5 + 1 = 6$ = 6(E)

Par la suite on cherche le minimum

Min (6, 1) = 2 donc le sommet B i.e

A	B	C	D	E	F	G	H	Sommet fixé
X	3 (A)	∞	∞	X	∞	∞	∞	E
X	2 (E)	∞	∞	X	6 (E)	∞	∞	B

Le point B est lié à A (déjà fait), aussi lié à E (déjà fait) on ne revient plus sur la liaison B $\longrightarrow$ A et B $\longrightarrow$ E donc on avance avec F et C et on met infini (∞) au reste des sommets correspondants

B $\xrightarrow{3}$ C : $\sigma = 3 + 2 = 5$ = 5(B)

E $\xrightarrow{1}$ F : $\sigma = 1 + 2 = 3$ = 3(B)

Par la suite on cherche le minimum : Min (5, 3) = 3 par suite le sommet **F** que je vais fixé

A	B	C	D	E	F	G	H	Sommet fixé
X	2 (E)	∞	∞	X	6(E)	∞	∞	B
X	X	5(B)	∞	X	3(B)	∞	∞	F

A	B	C	D	E	F	G	H	Sommet fixé
X	X	**5(A)**	∞	X	**3(A)**	∞	∞	**F**
X	X			X	X		∞	

On met infini à H et C, puis remplissons C, G et D

Par Suite,

F $\xrightarrow{1}$ C : $\sigma = 3 + 1 = 5$ = 4(F)

F $\xrightarrow{4}$ G : $\sigma = 4 + 3 = 7$ = 7(F)

F $\xrightarrow{2}$ D : $\sigma = 2 + 3 = 5$ = 5(F)

Par la suite on cherche le minimum : Min (5, 7, 4) = 4 donc nous allons fixer le point **C.**

A	B	C	D	E	F	G	H	Sommet fixé
X	X	**5(A)**	∞	X	**3(A)**	∞	∞	**F**
X	X	**4(F)**	**5(F)**	X	X	**7(F)**	∞	**C**

C'est lié à D et on ne peut plus liés à F car est déjà marqué par X.

Remarque

- Pour la colonne B : ∞ >3(A) >2(E) ;
- Pour la colonne C : ∞ >5(B) >4(F) ;
- Pour la colonne D : ∞ >5(F) ;
- Pour la colonne E : 1(A) > ∞ ;
- Pour la colonne F : ∞ >6(E)> 3(B) ;
- Pour la colonne G : ∞ >7(F).

Par conséquent dans la colonne G je ne peux mettre ∞ à la suite de 7(F) car il y aura **distorsion** donc je suis obligé de reporté **7(F)** ce que dit l'algorithme de **DIJKSTRA**.

A	B	C	D	E	F	G	H	Sommet fixé
X	X	**5(A)**	∞	X	**3(A)**	∞	∞	**F**
X	X	**4(F)**	**5(F)**	X	X	**7(F)**	∞	C
X	X	X	**5(F) / 8(C)**	X	X	**7(F)**	∞	

C $\xrightarrow{4}$ D : $\sigma = 4 + 4 = 8$ = 8(C) or 8(F) >5(F) **donc je reporté 5(F) le poids minimum donc fixons le point D i.e**

A	B	C	D	E	F	G	H	Sommet fixé
X	X	**5(A)**	∞	X	**3(A)**	∞	∞	**F**
X	X	**4(F)**	**5(F)**	X	X	**7(F)**	∞	C
X	X	X	**5(F)**	X	X	**7(F)**	∞	**D**

A	B	C	D	E	F	G	H	Sommet fixé
X	X	X	5(F)	X	X	7(F)	∞	**D**
X	X	X	X	X	X	6(D)	9(D)	**G**
X	X	X	X	X	X	X	8(G)	**H**

D $\xrightarrow{1}$ H : $\sigma = 1 + 5 = 6 = 6(D)$

D $\xrightarrow{4}$ G : $\sigma = 4 + 5 = 9 = 9(D)$

MIN (6,9)= 6 **donc on fixe G**

G $\xrightarrow{2}$ H : $\sigma = 2 + 6 = 8(D)$

A	B	C	D	E	F	G	H	Sommet fixé
0	∞	∞	∞	∞	∞	∞	∞	A
X	3 (A)	∞	∞	X	∞	∞	∞	E
X	**2 (E)**	∞	∞	X	**6 (E)**	∞	∞	**B**
X	X	**5(A)**	∞	X	**3(A)**	∞	∞	**F**
X	X	**4(F)**	**5(F)**	X	X	**7(F)**	∞	**C**
X	X	X	**5(F)**	X	X	**7(F)**	∞	**D**
X	X	X	**X**	X	X	**6(D)**	9(D)	**G**
X	X	X	**X**	X	X	**X**	8(G)	**H**

- Plus courte chaine entre A et H

<u>Principe en écrit en envers</u>. Donc on commence par H

Je pars dans la colonne de **H**, et dans la dernière ligne et prend la dernière lettre soit **G**. je vais dans **G** même chose **D**. Je vais dans D, mais la dernière c'est **F** je vais dans F c'est **B**, je vais dans maintenant c'est **E** et je vais dans E c'est **A** et je vais dans A il n'y a rien. Donc la plus courte chaine entre A et H est :

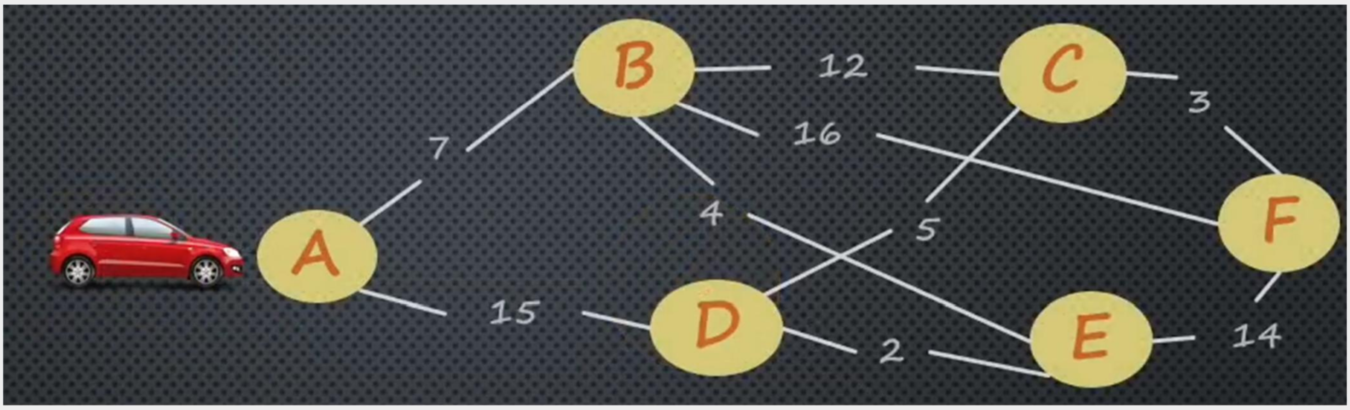
A-E-B-F-D-G-H

Plus courte chaine en A et C est :

A-E-B-F-C

Attention cet algorithme n'assure pas de l'unicité de la solution s'il ya plusieurs solutions. Les chaines peuvent avoir les longueurs différentes

Exercice 1. Considérons un ensemble de ville, de la ville A jusqu'à la vile F. La distance entre chaque ville est matérialisée par des nombres positive.



1) Déterminer le plus court chemin pour aller de A à F

2) Déduire la distance parcourue de la ville A à la ville B

3) Déterminer le trajet le plus optimal pour aller de A à C

Solution

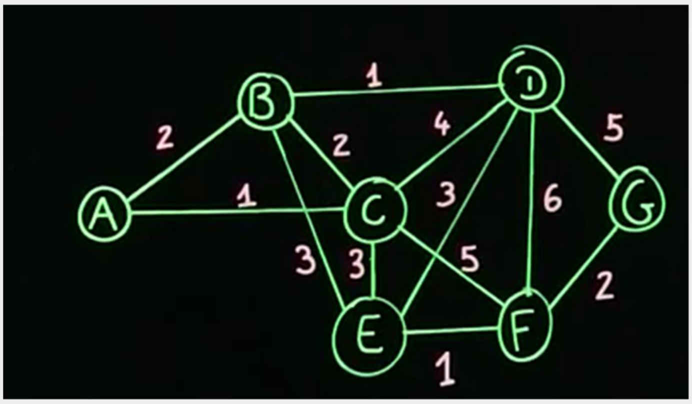
	∞	∞	∞	∞	∞	A
A	B	C ∞	D	E ∞	F ∞	B
0A			∞			E
0A	7A	∞	15A			D
•	7A	19B		11B	∞23B	C
•	•		13E	11B	25E	F
•	•	18D	13E	•		F
•	•	18D	•	•	21C	
•	•	•	•	•	21C	
•	•	•	•	•	•	

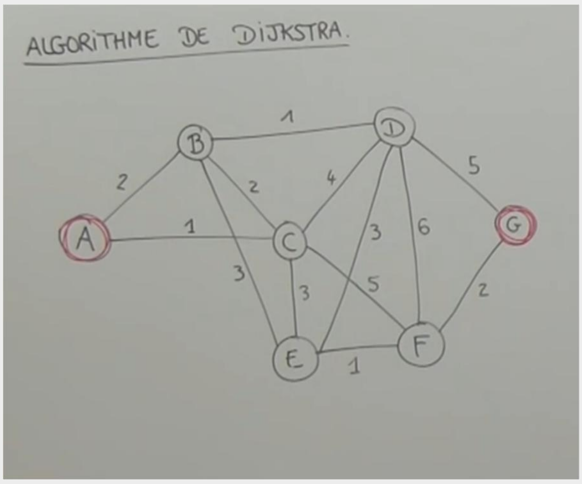
1. le plus court chemin pour aller de A à F est :

A-B-E-D-C-F

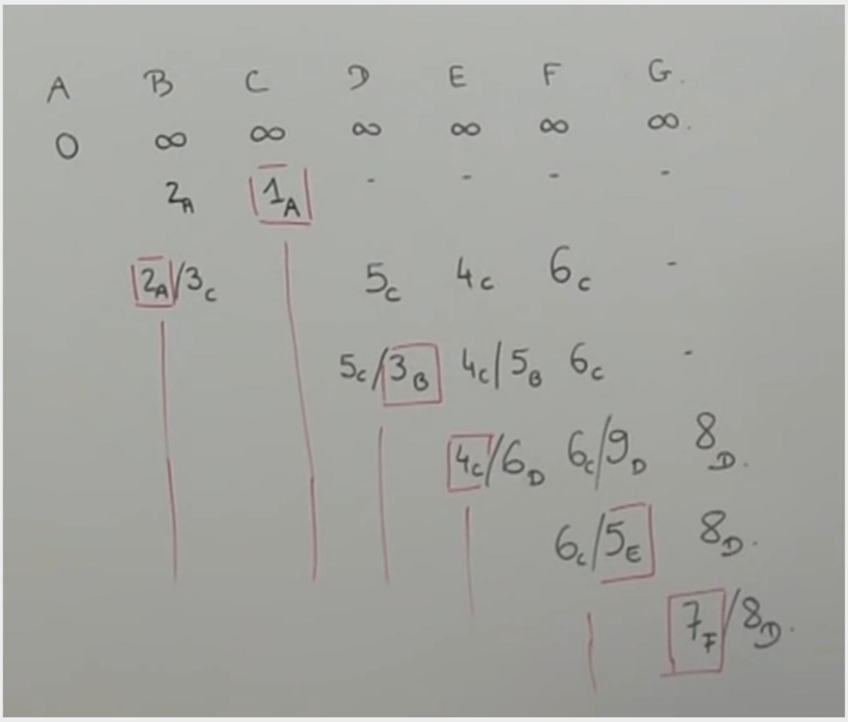
2. Déduire la distance parcourue de la ville A à la ville B est : 21 km
3. le trajet le plus optimal pour aller de A à C est : A -B -E-D-C

Exercice 2. Soit le graphe ci-dessous

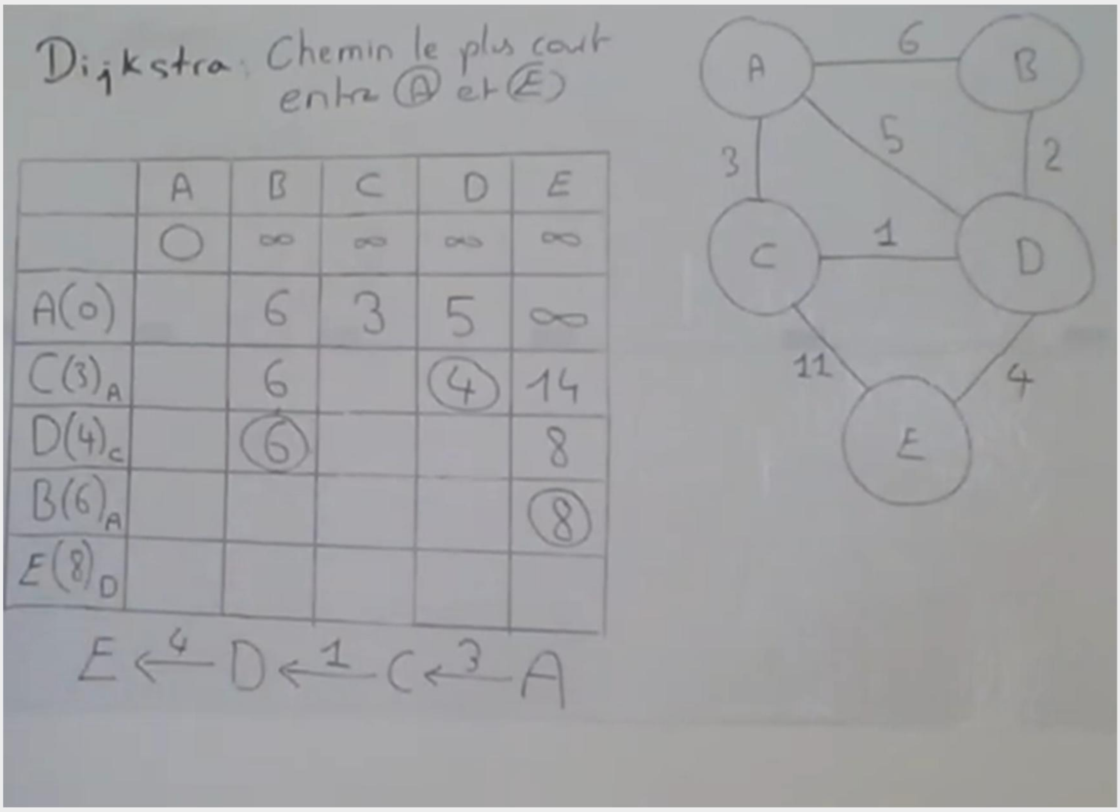




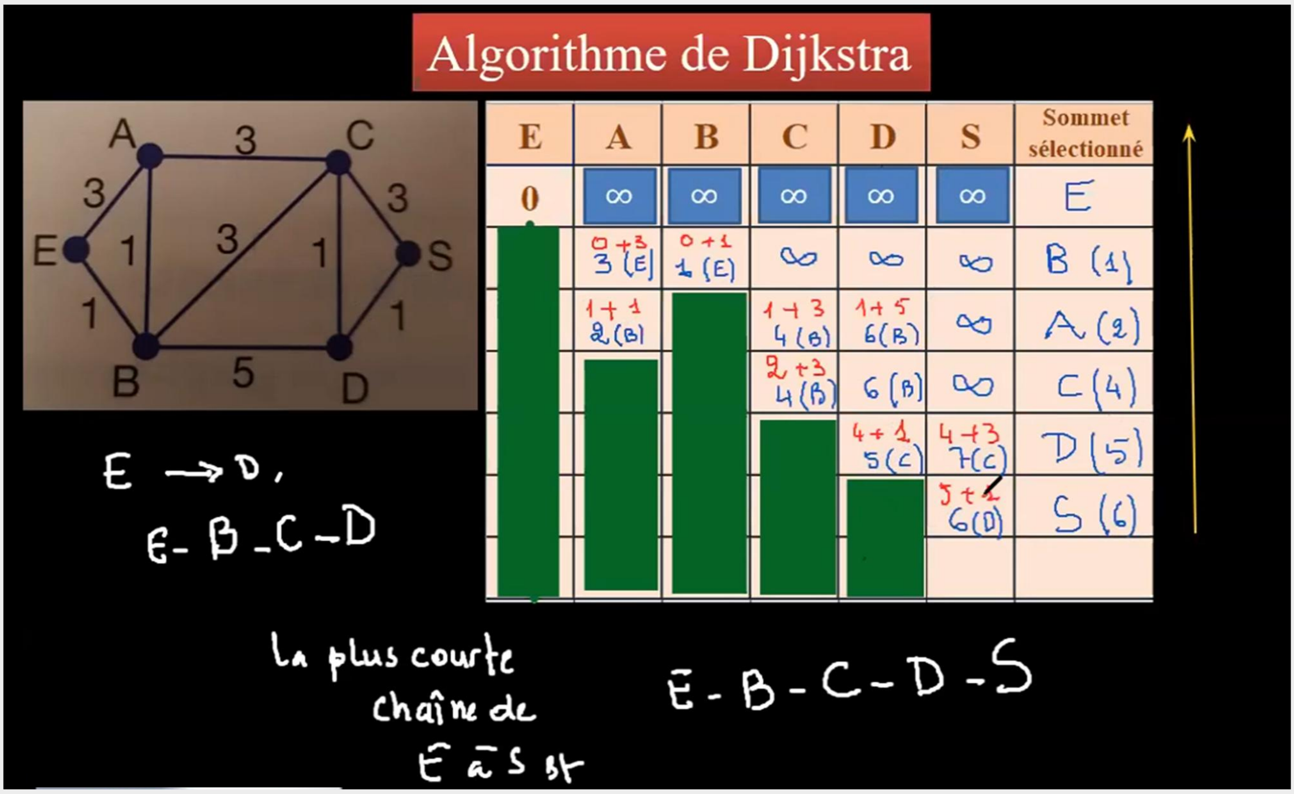
1) Déterminer la plus courte chaine entre A et G graphe



Donc A-C-E-F-F-G ; 7 minutes



Exercice corrigé

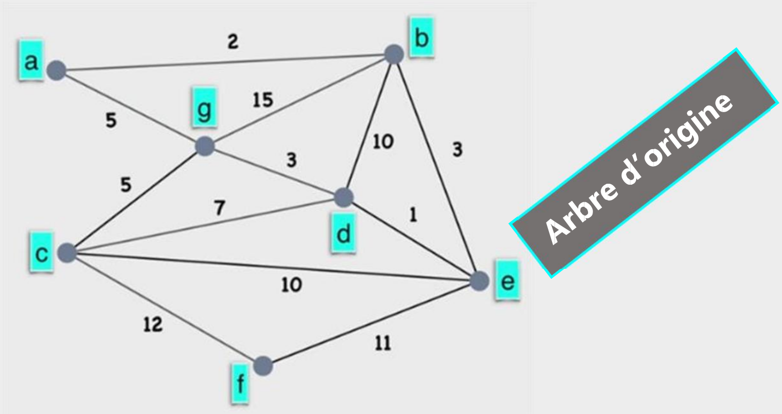


III. Arbres et Arbre couvrants

1. Arbres

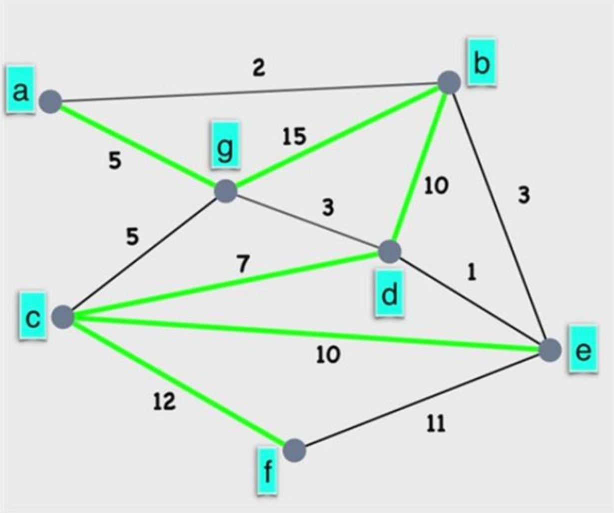
2. Arbre couvrant de poids minimum

Soit un graphe connexe pondéré (à chaque arête un poids positif) et sur chaque sommets les étiquettes a, b, c, d, e, f et g.

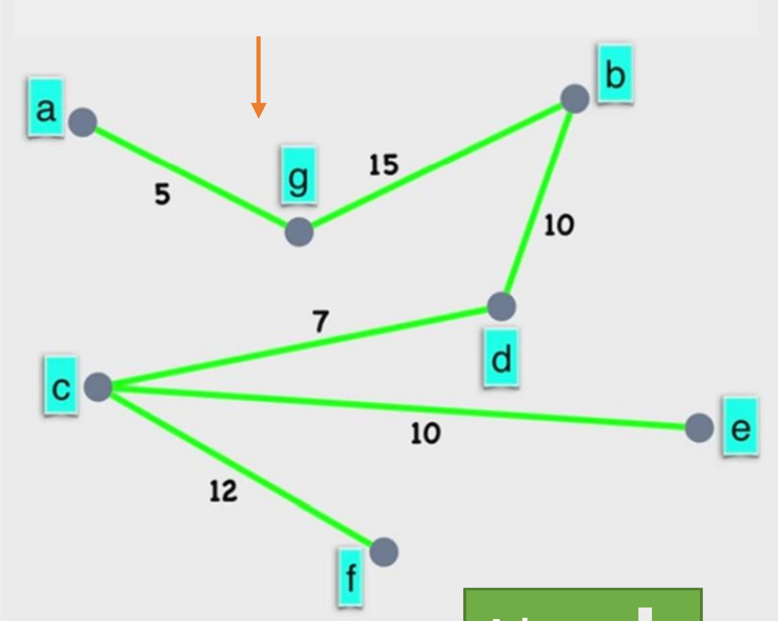


On veut extraire un arbre contenant tous les sommet (arbre couvrant) de poids minimum.

Exemple : Un graphe qui satisfait tous nos conditions



Ensuite oublions les arêtes qui sont non verte.



Arbre vert

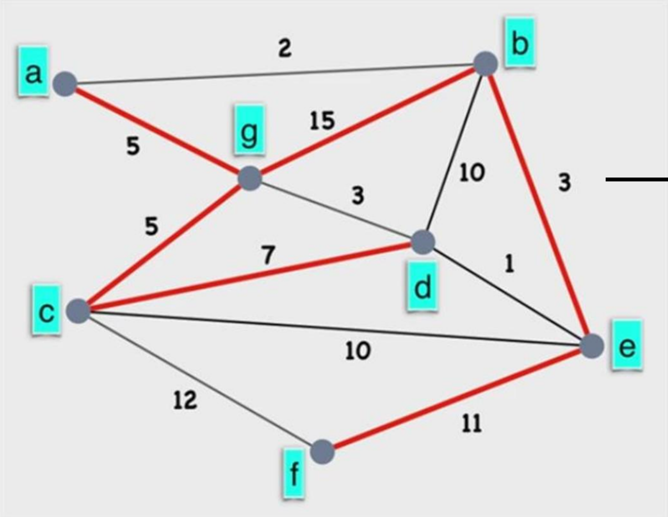
On peut constater facilement que sa c'est un **arbre contenant tous les sommets** d'origine. Par contre, si je calcule le poids :

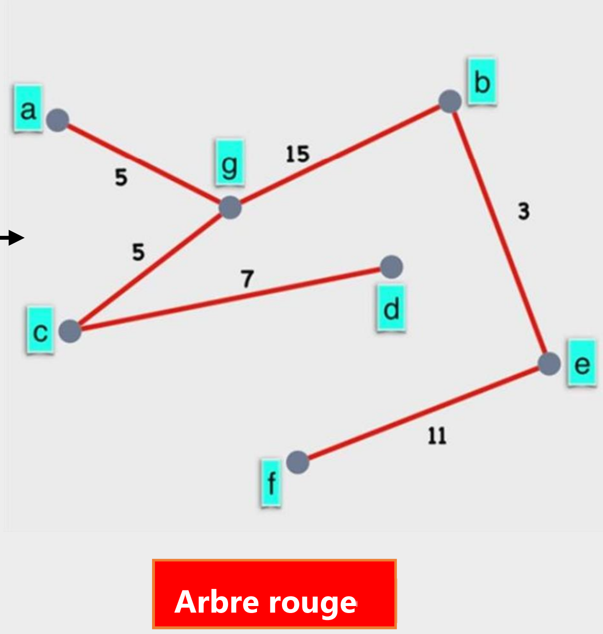
Poids(T) = 5+15+10+7+10+12

Poids(T) = 59.

Est-il de poids minimal ? Non ! pour ça, il suffit de prendre un arbre d'arêtes rouges. Nous constatons que c'est un arbre extrait du graphe l'origine.

Illustrations :





Par suite **poids(T)**=5+15+5+7+3+11 =46 < **Poids(T) = 59.**

Est-ce que l'arbre est de poids minimal ? Non !

3. Algorithme de Prim

Algorithme de Prim

Entrée : un graphe pondéré

Sortie: un arbre

Début:

1. Trouvez une arête avec de poids minimum dans le graphe. L'assombrir et encerclez ses deux sommets.

2. Trouvez une arête avec de poids minimum dans le graphe. L'assombrir et encerclez ses deux sommets.

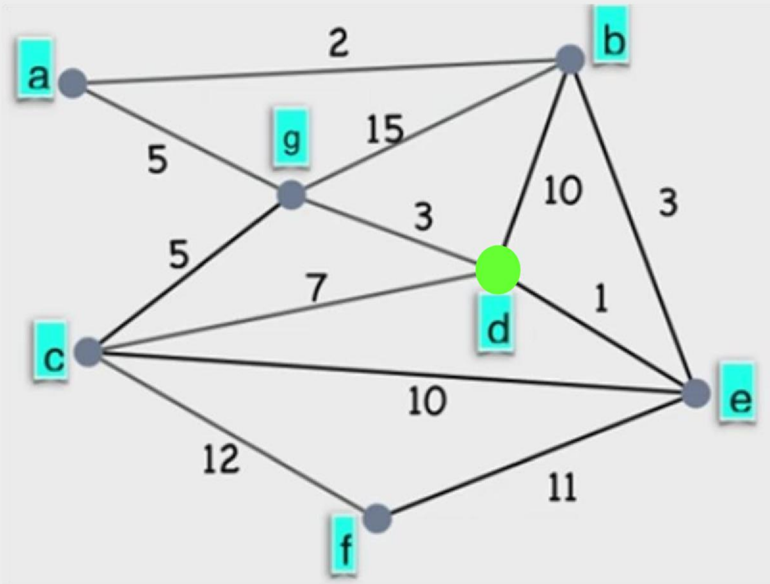
3. Trouvez une arête de poids minimum parmi les arêtes non assombris restants ayant un sommet encerclé et un sommet non encerclé. Assombrir cette arête et son sommet non encerclé.

4. Répétez l'étape 2 jusqu'à ce que tous les sommets soient encerclés.

Fin

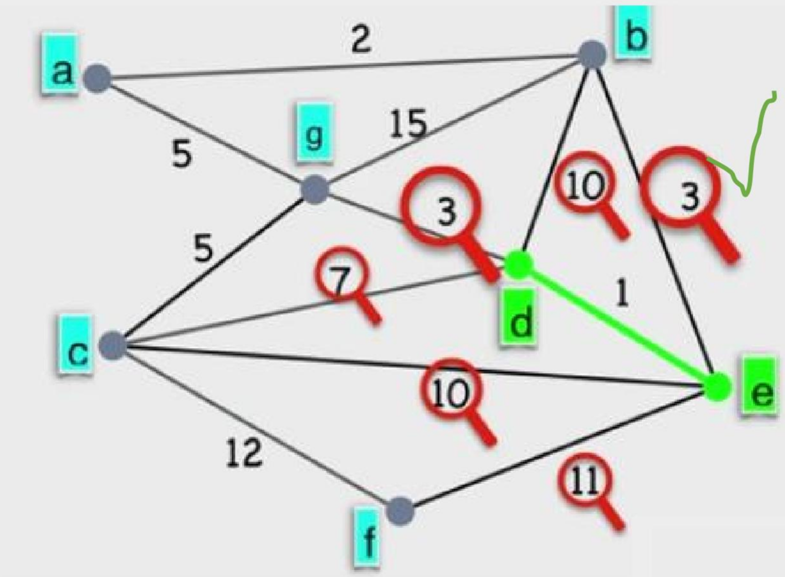
Application :

Prenons le graphe d'origine



L'algorithme de Prim, il dit :

1er étape : Considéré n'importe quel sommet de départ. Pour ce graphe, je pars du sommet d et ce sommet il peut être vu comme une sorte d'arbre partiel, qui ne couvre pas encore tous le graphe mais cet arbre on doit le faire grossir. À chaque étape on doit lui ajouter un sommet une arête, l'arête qui va être choisir serais celle qui est de poids minimal qui sort de l'arbre.

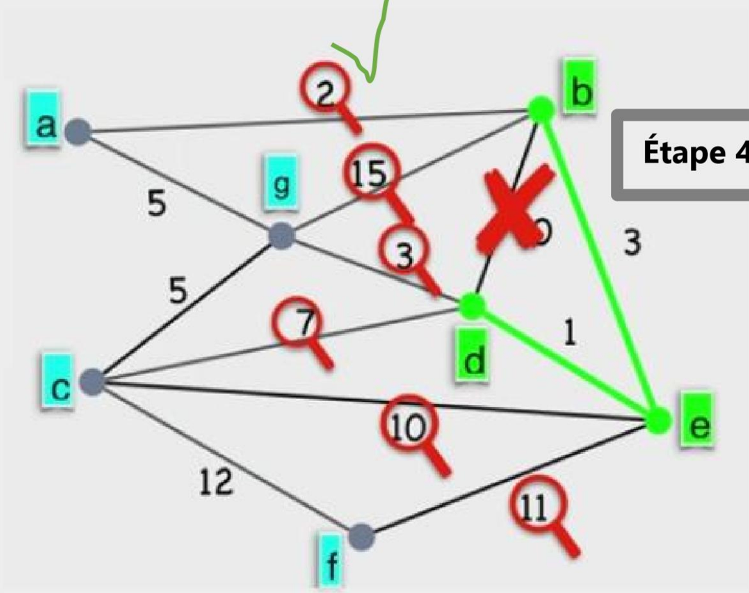


L'algorithme de Prim, il dit :

2^{ième} étape : choisir l'arête de poids minimale qui sort de l'arbre, dont l'arête **d-e** de poids 1, et appliquons la même règle qu'à la 1er étape i.e on choisir les arêtes hors de l'arbre (les poids 3, 10, 3, 7, 10 et 11) et nous avons deux poids minimal identique, la loi de Prim dit : choisir, soit l'arête e-b ou d-g. Choisir e-b

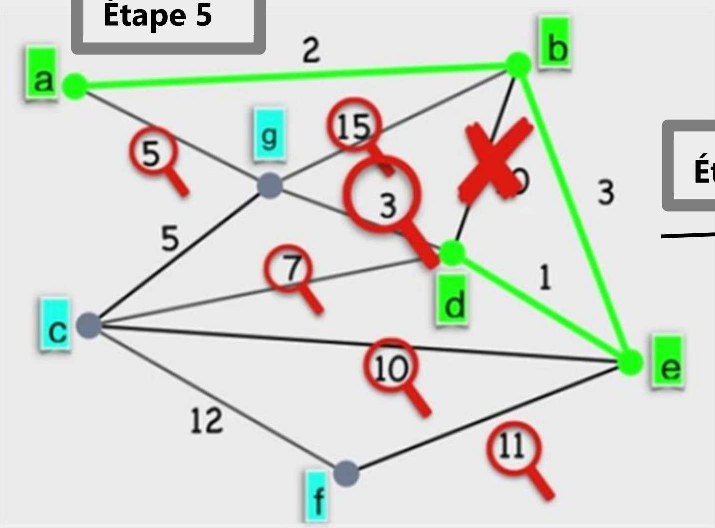
Continuions sur les mêmes règles ou les poids d'arête candidate sont : 2, 15, 3, 7, 10 et 11.

Attention l'arête entre d et b n'est pas une arrête candidate, puisqu'elle a ses deux extrémités dans l'arbre vert.

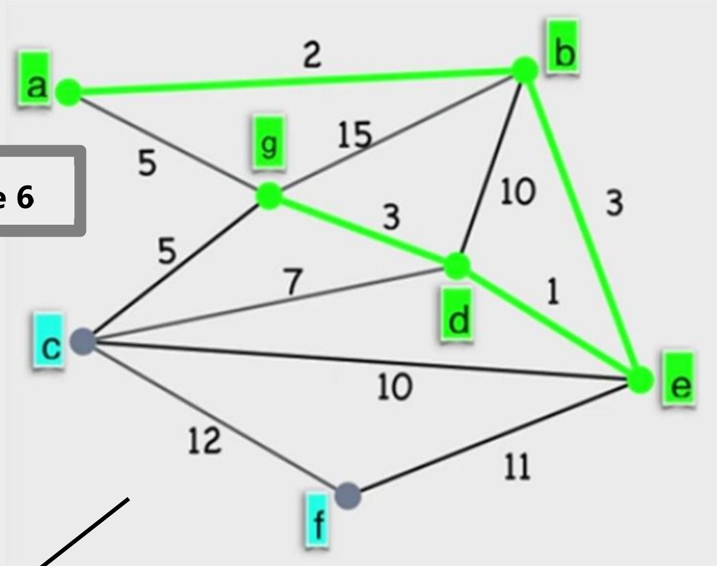


Étape 4

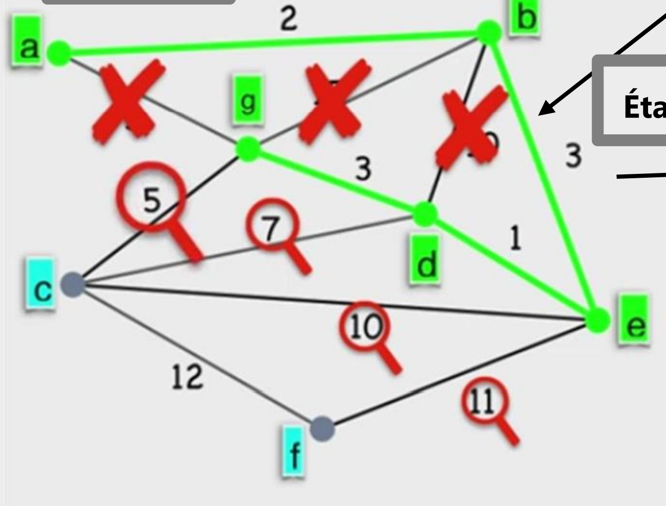
Étape 5



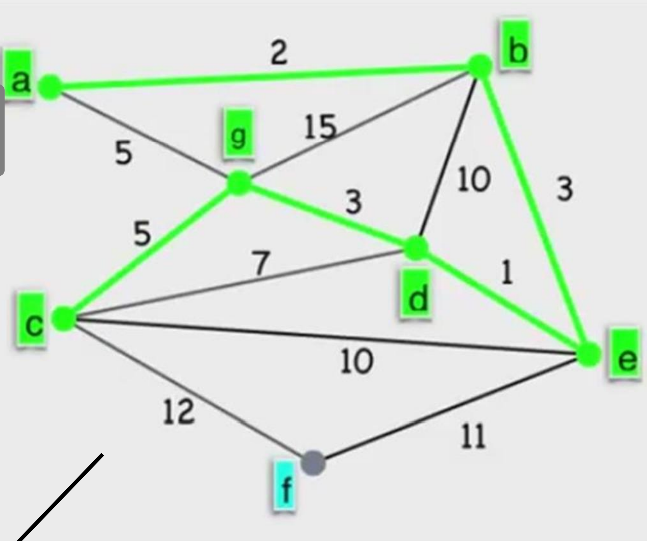
Étape 6



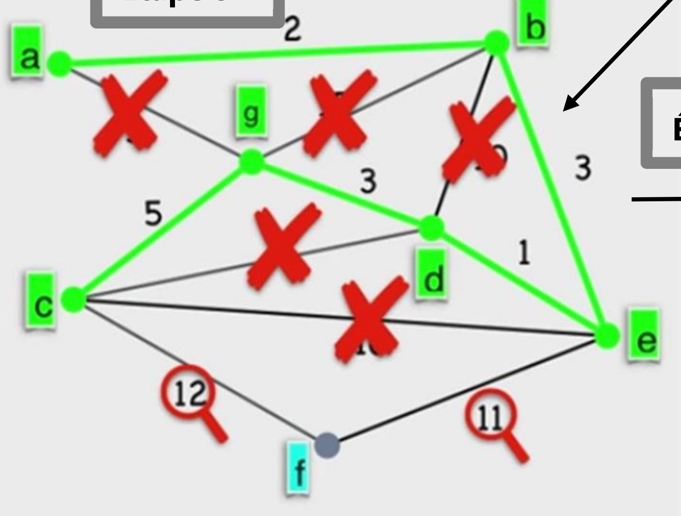
Étape 7



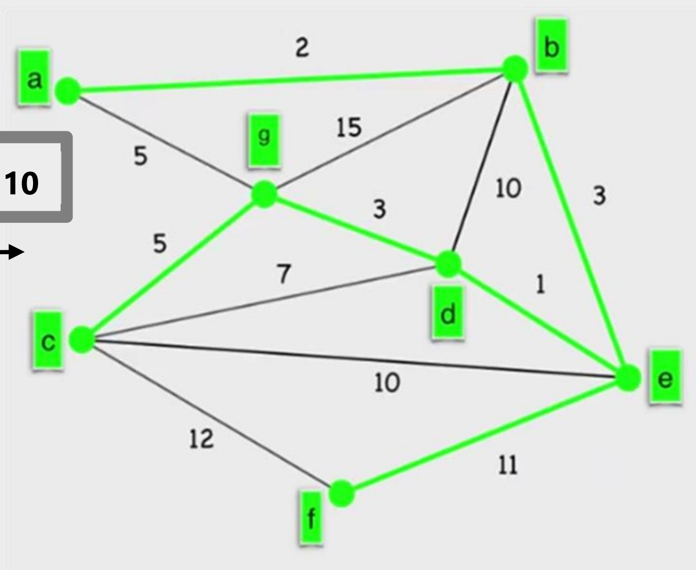
Étape 8



Étape 9



Étape 10



En fin, nous avons un arbre contenant tous les sommets du graphe d'origine

Poids(T) = 2+3+4+3+3+11

Poids(T) = 25. Donc on a la garantie que cet arbre est de **poids minimal.**