Théorie des graphes

Chapitre 4: Arbre couvrant à Poids minimal

asma.ghdami@esprit.tn

October 19, 2022

Outline

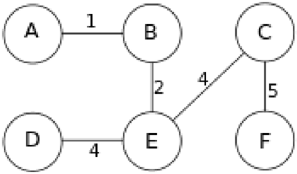
Arbre couvrant: définitions et propriétés

Algorithme de KrusKal (1956)

Algorithme de Prim (1957)

Définition: Arbre

- Un arbre est une famille particulière du graphe.
- Un arbre A est un graphe tel que: A est connexe et A ne contient pas de cycles.

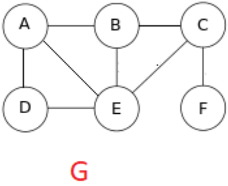


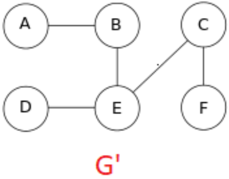
Rappel: Graphe connexe

Un graphe G est dit connexe si pour tout couple de noeud (i, j) il existe un chemin reliant i et j.

Rappel: Graphe partiel

Un graphe partiel $G' = (V, E')$ d'un graphe $G = (V, E)$ est un graphe qui a le même ensemble de sommet que G et l'ensemble des arêtes $E' \subset E$.





Rappel: Graphe connexe

Un graphe G est dit connexe si pour tout couple de noeud (i, j) il existe un chemin reliant i et j.

Rappel: Graphe partiel

Un graphe partiel $G' = (V, E')$ d'un graphe $G = (V, E)$ est un graphe qui a le même ensemble de sommet que G et l'ensemble des arêtes $E' \subset E$.

Arbres couvrants

Un arbre couvrant d'un graphe *G* est un graphe partiel connexe sans cycle.

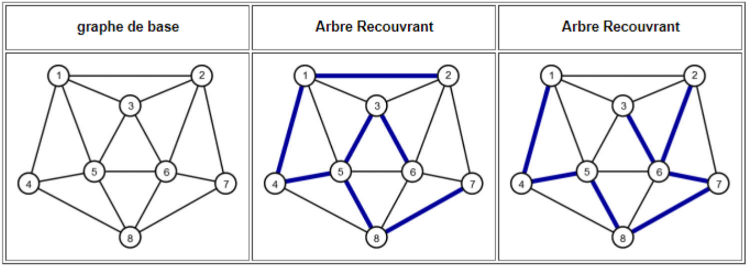


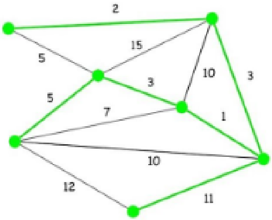
Figure : Un graphe connexe non valué et deux arbres recouvrants possibles.

Poids d'un graphe

Le poids (ou coût) d'un graphe est la somme des poids des arêtes du graphe.

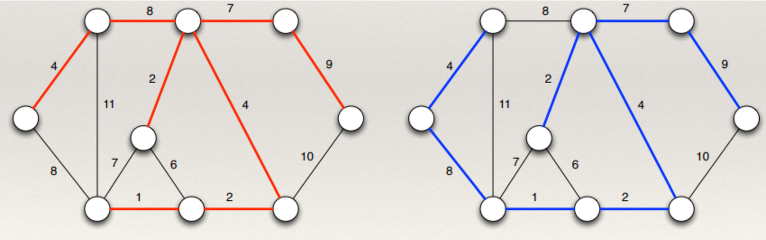
Arbre couvrant à poids minimum

Soit un graphe non orienté, valué *G*. On appelle arbre couvrant de poids minimum de G (noté ACPM), tout arbre couvrant dont la somme des poids des arêtes le constituant est minimal.



Remarque

L'arbre couvrant à poids minimal n'est pas forcément unique.



Propriétés

Soit G un graphe à "n" sommets et "m" arêtes. Les propriétés suivant sont équivalentes:

• G connexe sans cycles.
• G connexe et $m = n - 1$.
• Entre 2 sommets il existe un unique chemin.
• Un graphe non connexe n'a aucun arbre recouvrant.
• Un graphe connexe a forcément (au moins) un arbre couvrant (par exemple un arbre de parcours).

Content

Algorithme de KrusKal (1956)

Idée générale

• On part d'une forêt (un ensemble d'arbres est appelé une forêt) d'arbres constitués de chacun des sommets isolés du graphe.
• À chaque itération, on ajoute à cette forêt l'arête de poids le plus faible ne créant pas de cycle avec les arêtes déjà choisies.
• On stoppe quand on a examiné toutes les arêtes.

On définit par "T" l'arbre couvrant, "m" le nombre de liens qui composent T et "n" le nombre de sommets du graphe G.

Algoritme de KrusKal

Etape (0) : Initialisation

$T = \phi$

Ordonner les arêtes suivant des poids croissants

On suppose : $P(a_1) \leqslant P(a_2) \leqslant \ldots \leqslant P(a_n)$. (P(a) est le poids de l'arête

Etape (1) :

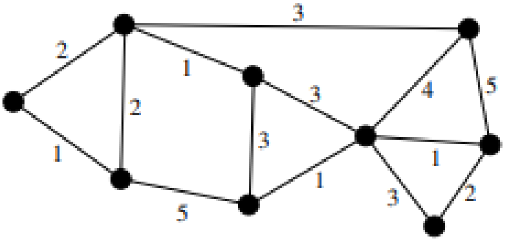
Pour $k : 1, \ldots n$

Tant Que : $m < n - 1$

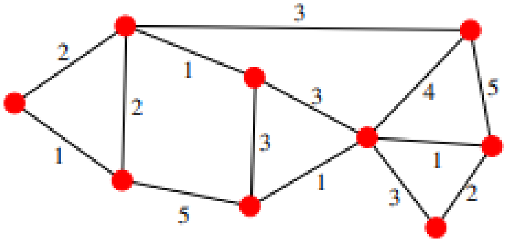
Faire : $T = T \cup \{a_k\}$, $m \leftarrow m + 1$ si $T \cup \{a_k\}$ ne contient pas de circuits.

Fin TantQue

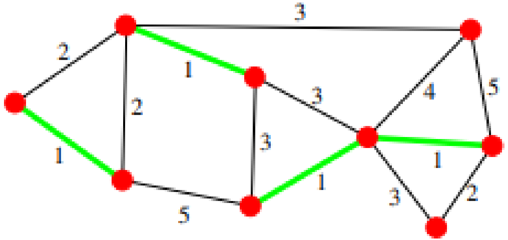
Exemple: Algorithme de Kruskal



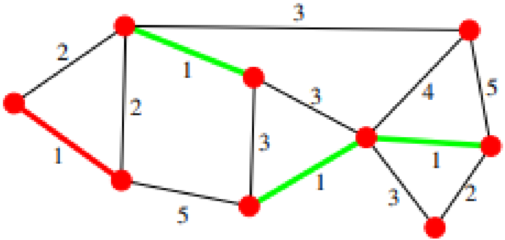
Exemple: Algorithme de Kruskal

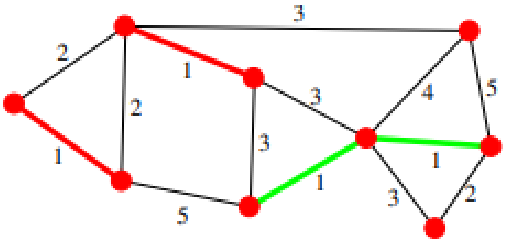


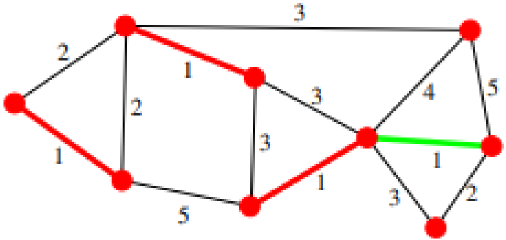
Exemple: Algorithme de Kruskal

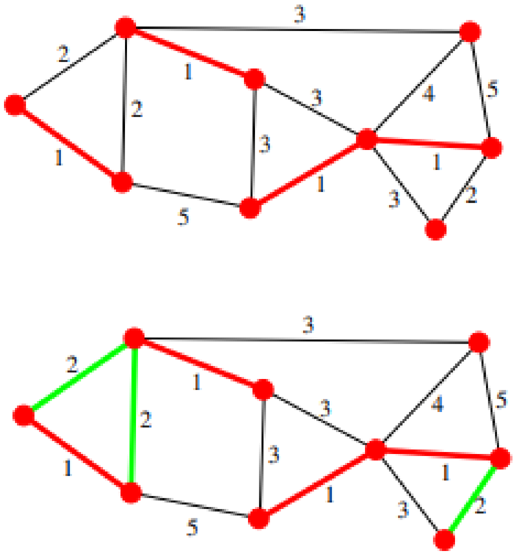


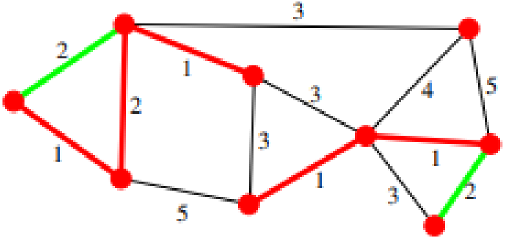
Exemple: Algorithme de Kruskal

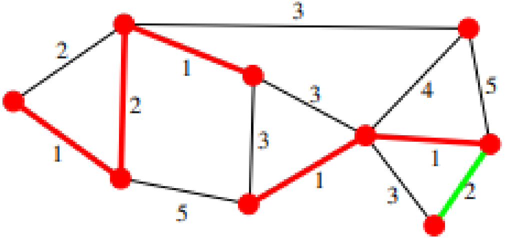


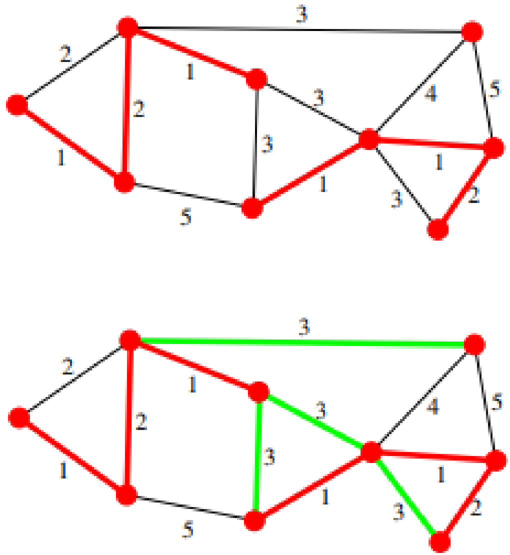

3
2
1
2
3
4
5
3
1
1
1
3
2
5

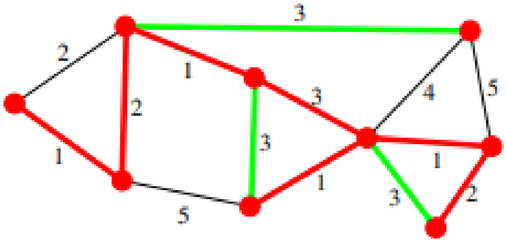

3
2
1
2
3
4
5
3
1
1
1
3
2
5

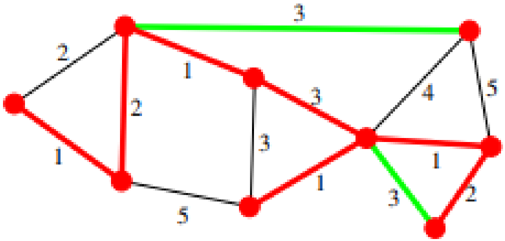

3
2
1
3
4
5
2
3
1
1
1
3
2
5
3
2
1
3
4
5
2
3
1
1
1
3
2
5

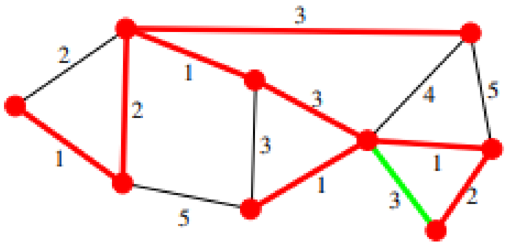

3
2
1
5
3
4
2
3
1
1
1
3
2
5

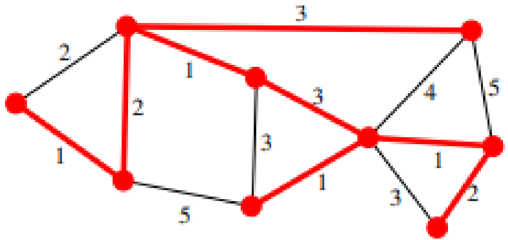

3
2
1
5
3
4
2
3
1
1
1
3
2
5

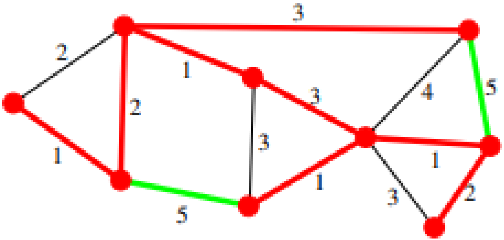

3
2
1
3
4
5
2
3
1
1
1
3
2
5
3
2
1
3
4
5
2
3
1
1
1
3
2
5

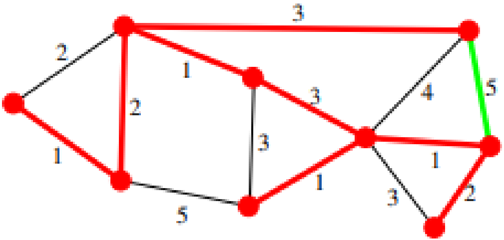

3
2
1
2
3
4
5
3
1
1
1
3
2
5

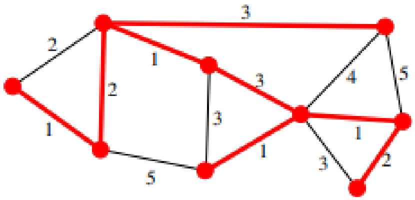

3
2
1
2
3
4
5
3
1
1
1
3
2
5


3
2
1
3
4
5
2
3
1
1
1
3
2
5


3
2
1
3
4
5
2
3
1
1
1
3
2
5

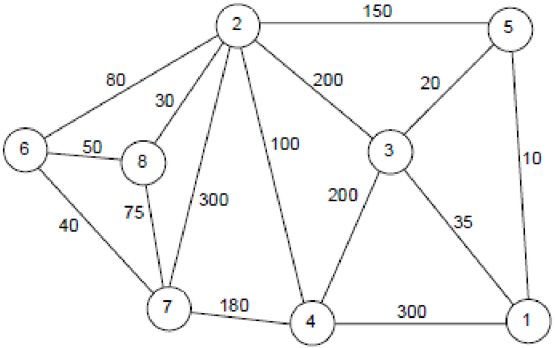

3
2
1
2
3
4
5
3
1
1
1
3
2
5


3
2
1
2
3
4
5
3
1
1
1
3
2
5



$$1 + 1 + 1 + 1 + 2 + 2 + 3 + 3 = 14$$

Exemple 2: Algorithme de Kruskal



Correction exemple 2: Algorithme de Kruskal

Tri des arêtes par poids croissant :

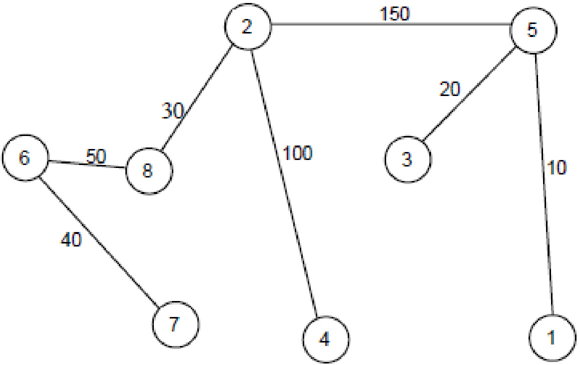
arête	[1,5]	[3,5]	[2,8]	[3,1]	[6,7]	[6,8]	[7,8]	[2,6]	[2,4]	[2,5]	[4,7]	[3,4]	[2,3]	[1,4]	[2,7]
poids	10	20	30	35	40	50	75	80	100	150	180	200	200	300	300

D'après l'algorithme :

On peut choisir [1,5] puis [3,5] puis [2,8] mais pas [3,1] (qui formerait un cycle avec [1,5] et [3,5]).
Puis on peut choisir [6,7] puis [6,8] mais pas [7,8] ni [2,6].
Puis on peut choisir [2,4] puis [2,5].

On s'arrête ici car le nombre d'arêtes choisies est 7 (c'est à dire n-1, puisque n = 8).

Correction exemple 2: Algorithme de Kruskal

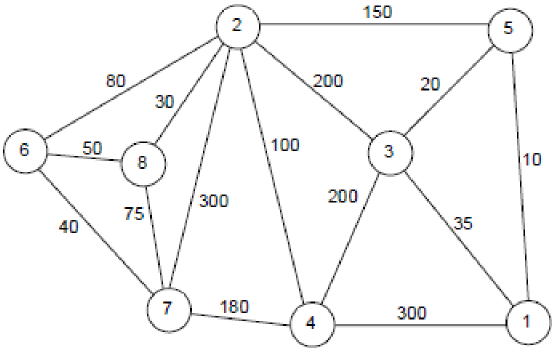


L'arbre obtenu est de poids 400.

Arbre Couvrant Maximal: Algorithme de Kruskal

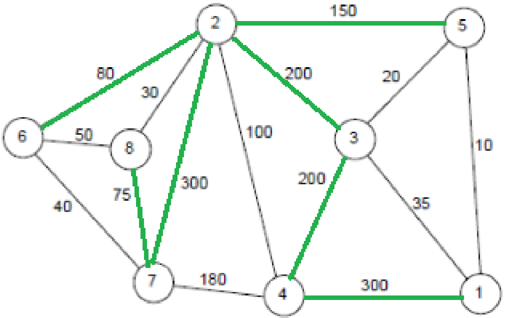
- L'arbre couvrant maximal est l'arbre couvrant rendant maximal la somme des similitudes de ses arêtes.
- On peut le former en s'inspirant de l'algorithme de Kruskal.
- On classe les arêtes par similitude décroissante. L'arbre est initialement formé de l'arête présentant la plus grande similitude. Jusqu'à ce que tous les sommets lui appartiennent, on lui ajoute à chaque étape l'arête de plus grande similitude unissant un sommet couvert par l'arbre déjà formé à un sommet non encore couvert.

Exemple 2: Algorithme de Kruskal



Déterminer l'arbre Couvrant Maximal.

Exemple 2: Algorithme de Kruskal



Content

Arbre couvrant: définitions et propriétés

Algorithme de KrusKal (1956)

Algorithme de Prim (1957)

Algorithme de Prim (1957)

Définition

Dans un graphe $G = (V, E)$, le co-cycle d'un ensemble de sommets V' est l'ensemble des arêtes (u, v) telles que $u \in V'$ et $v \in V \setminus V'$.

Principe

• Au départ un sommet x_i est choisi arbitrairement, ce sommet constitue l'arbre générateur T de poids minimal.
• Parmi toutes les arêtes incidentes à x_i, choisir celle de poids minimal: (x_i, x_j). Le nouveau arbre obtenu est constitué des sommets x_i et x_j et de l'arête (x_i, x_j).
• Tant qu'il reste des sommets en dehors de l'arbre T, la nouvelle arête à introduire dans T correspond à l'élément du co-cycle de T de poids minimal.

Algorithme de Prim (1957)

Définition

Dans un graphe $G = (V, E)$, le co-cycle d'un ensemble de sommets V' est l'ensemble des arêtes (u, v) telles que $u \in V'$ et $v \in V \setminus V'$.

Principe

• Au départ un sommet x_i est choisi arbitrairement, ce sommet constitue l'arbre générateur T de poids minimal.
• Parmi toutes les arêtes incidentes à x_i, choisir celle de poids minimal: (x_i, x_j). Le nouveau arbre obtenu est constitué des sommets x_i et x_j et de l'arête (x_i, x_j).
• Tant qu'il reste des sommets en dehors de l'arbre T, la nouvelle arête à introduire dans T correspond à l'élément du co-cycle de T de poids minimal.

Algorithme de Prim

X l'ensemble des sommets du graphe.

$T = (X_T, A_T)$ l'arbre générateur (Vide au départ)

Ajouter x_1 à X_T

CC$\longleftarrow$co-cycle de X_T

Tant Que $|X_T| < |X|$, ($|X|$: *le* nombre des sommets de X). Faire :

(x, t) arête de CC de poids minimum.

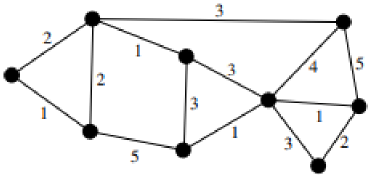
Ajouter (t) à X_T

Ajouter (x, t) à A_T

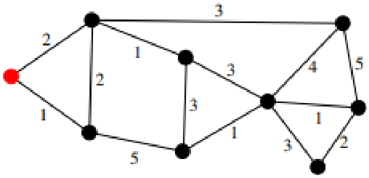
CC$\longleftarrow$ mise à jour du co-cycle de X_T

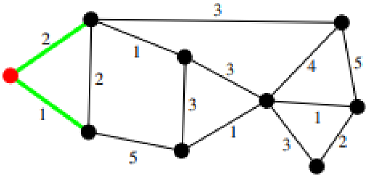
Fin Tant Que.

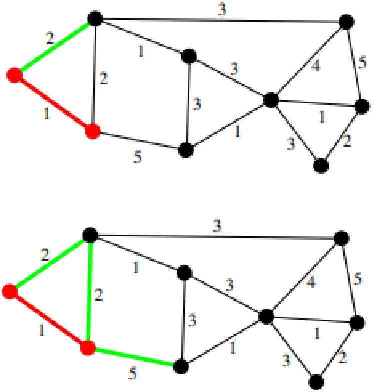
Exemple



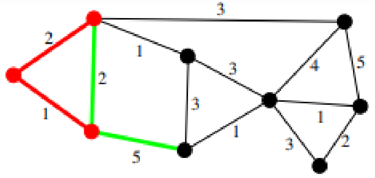
Exemple

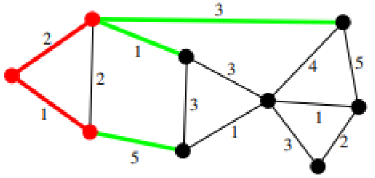




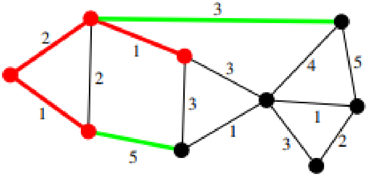

3
2
1
3
4
5
2
3
1
1
3
2
1
5
3
2
1
3
4
5
2
3
1
1
3
2
1
5

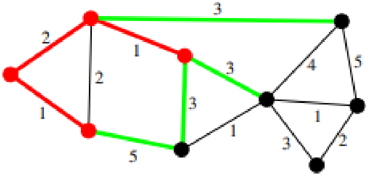
Exemple



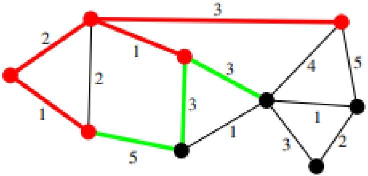


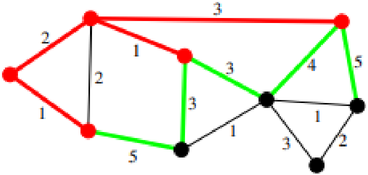
Exemple



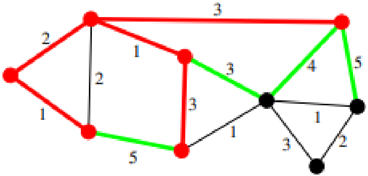


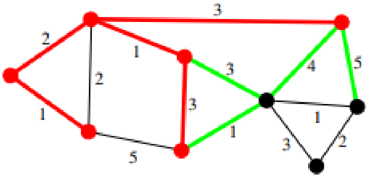
Exemple



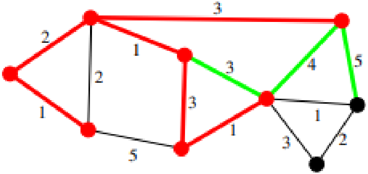


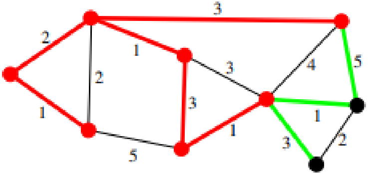
Exemple



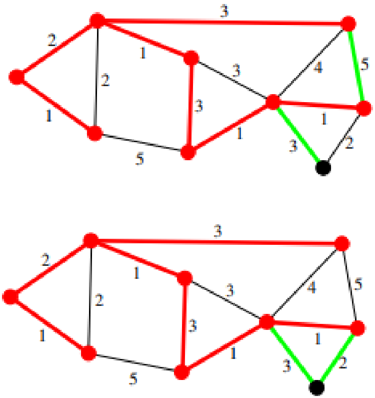


Exemple

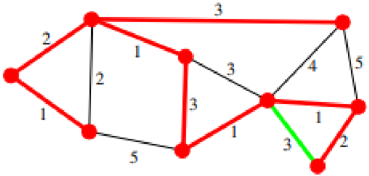


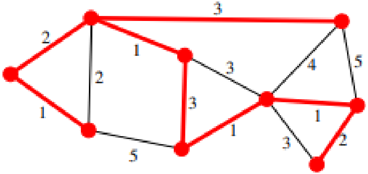


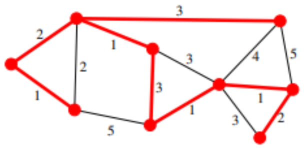
Exemple



Exemple



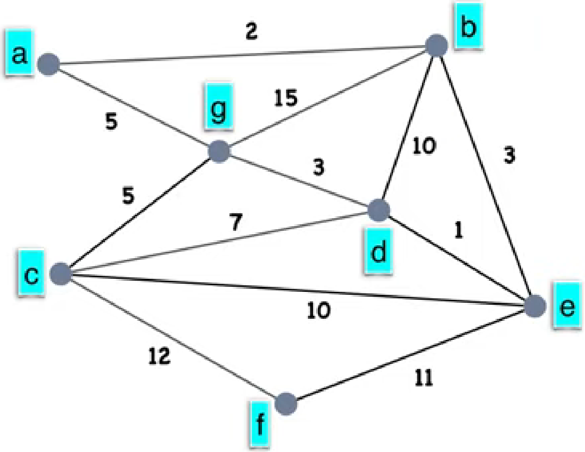




$1 + 2 + 1 + 3 + 3 + 1 + 1 + 2 = 14$

On obtient un arbre recouvrant différent mais de même poids

Exemple 2



Exemple 2: Correction