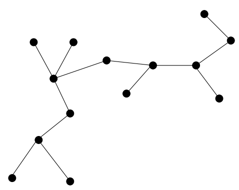
Université Clermont Auvergne
BUT Info 1A
R207 - Graphes
Année 2021-2022

Florent Foucaud
Malika More
Thibault Ralet
Carine Simon
Thierry Trévisan

TD2 - Arbres couvrants de poids minimal

1 Rappels de cours

1.1 Arbres



Construction Récursive

- Un graphe avec un seul sommet et zéro arête est un arbre.
- À partir d'un arbre avec au moins 1 sommet, en accrochant une feuille à n'importe quel sommet, on obtient un arbre avec un sommet et une arête de plus.

Remarques

- Un arbre à n sommets possède exactement $n-1$ arête.
- Un arbre avec au moins deux sommets a forcément au moins une feuille.

1.2 Arbre couvrant

Définition.

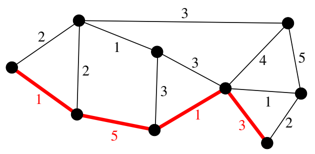
- graphe non orienté G
- arbre T $\begin{cases} \text{sommets : tous les sommets de } G \\ \text{arêtes : certaines arêtes de } G \end{cases}$

Remarques.

1. Un graphe peut avoir plusieurs arbres couvrants.
2. Un graphe non connexe n'a aucun arbre couvrant.
   (Autrement dit, un graphe qui a un arbre couvrant est forcément connexe.)
3. Un arbre n'a qu'un seul arbre couvrant, lui-même.
4. Un graphe connexe a forcément (au moins) un arbre couvrant.

1.3 Graphe valué

Définition (Graphe valué). Chaque arête a un *poids* (> 0).
Le poids d'une chaîne/d'un arbre est la somme des poids des arêtes qui la/le composent.



$$1 + 5 + 1 + 3 = 10$$

Question étant donné un graphe valué connexe, on veut construire un arbre couvrant de poids total le plus petit possible.

1.4 Algorithme de Prim

Méthode

- Initialiser T avec $\begin{cases} \texttt{sommets : un sommet de } G \texttt{ qu'on choisit} \\ \texttt{arêtes : aucune} \end{cases}$
- Répéter :
  - ⋆ Trouver toutes les arêtes de G qui relient un sommet de T et un sommet extérieur à T
  - ⋆ Parmi celles-ci, choisir une arête de poids le plus petit possible
  - ⋆ Ajouter à T cette arête et le sommet correspondant
- S'arrêter dès que tous les sommets de G sont dans T
- Retourner T

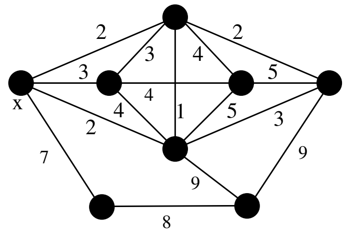
1.5 Algorithme de Kruskal

Méthode

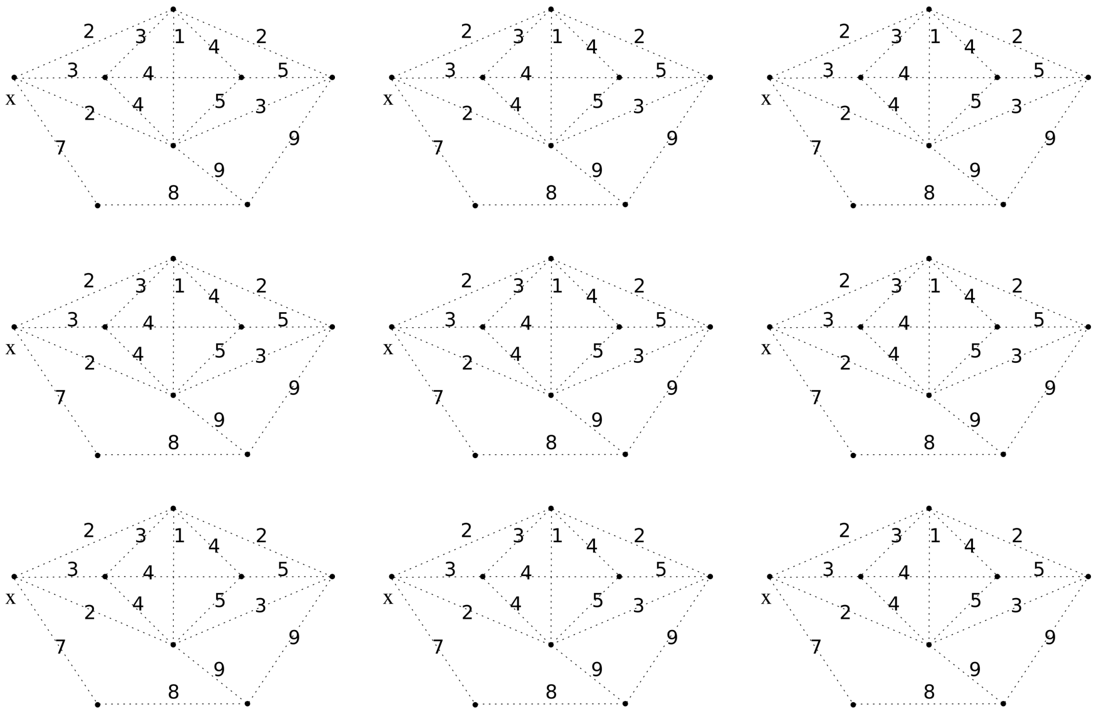
- Initialiser T avec $\begin{cases} \texttt{sommets : tous les sommets de } G \\ \texttt{arêtes : aucune} \end{cases}$
- Traiter les arêtes de G l'une après l'autre par poids croissant :
  - ∗ Si une arête permet de connecter deux composantes connexes de T,
    - ⋆ alors l'ajouter à T
    - ⋆ sinon ne rien faire
  - ∗ Passer à l'arête suivante
- S'arrêter dès que T est connexe
- Retourner T

2 Exercices de mise en œuvre des algorithmes

Exercice 1. On désire calculer un arbre couvrant de poids minimum pour le graphe G suivant.

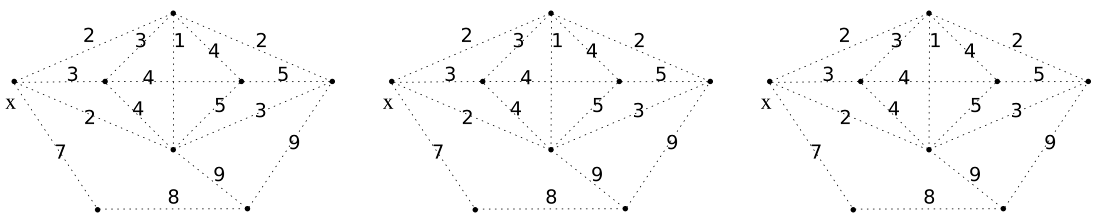


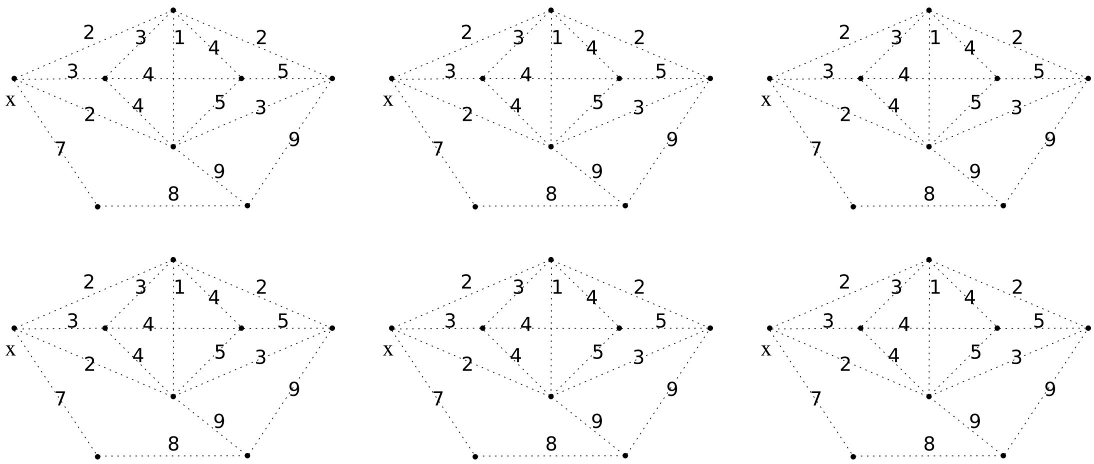
1. Appliquer l'algorithme de Kruskal au graphe G (détaillez bien chaque étape en utilisant le canevas ci-dessous).



   L'arbre couvrant a pour poids :

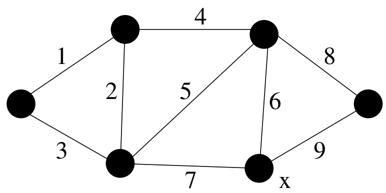
2. Appliquer l'algorithme de Prim au graphe G en commençant par le sommet x (détaillez bien chaque étape en utilisant le canevas ci-dessous).



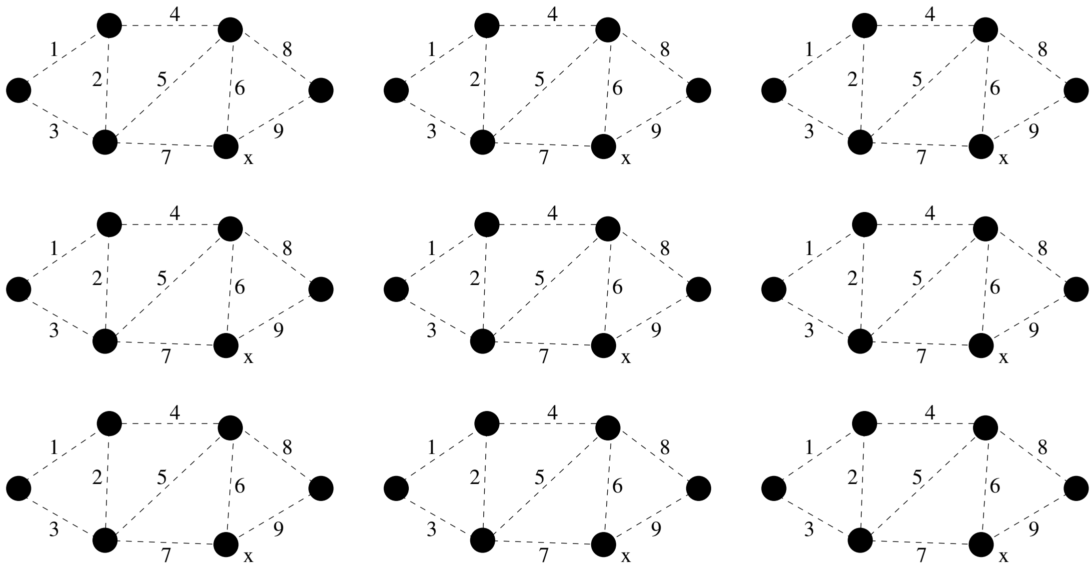


L'arbre couvrant a pour poids :

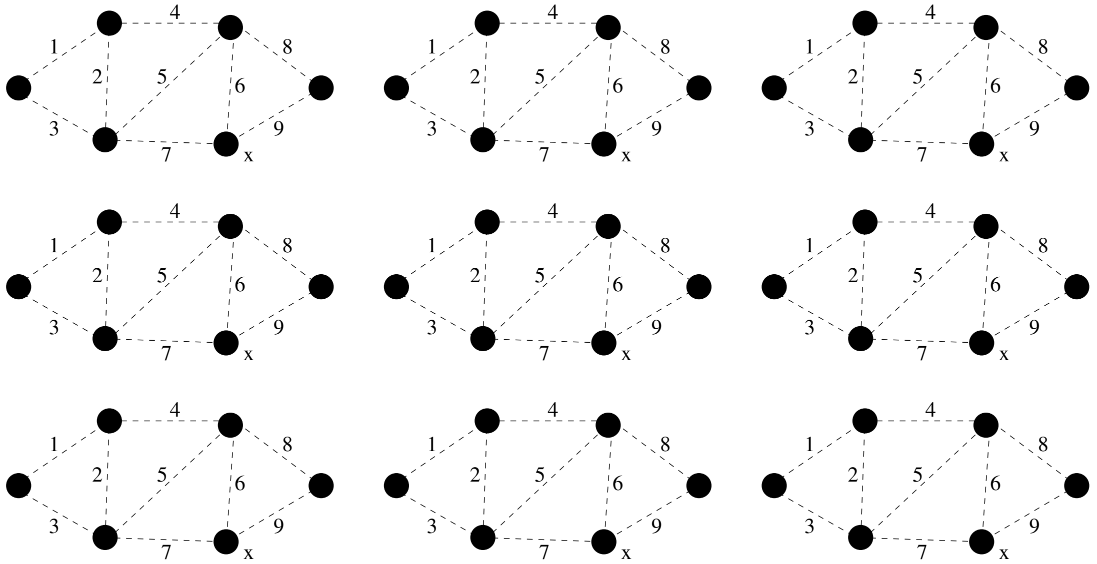
Exercice 2. Soit G_4 le graphe valué suivant :



1. Appliquer l'algorithme de Kruskal au graphe G_4. (détaillez bien chaque étape en utilisant le canevas ci-dessous)

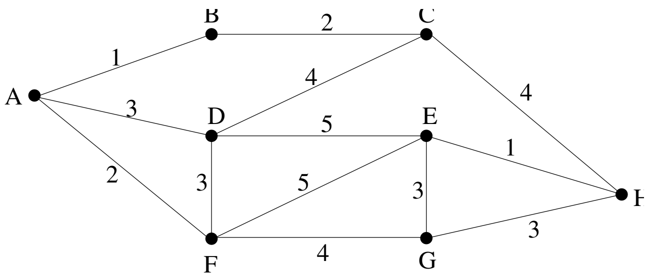


2. Appliquer l'algorithme de Prim au graphe G_4 : on commencera par le sommet x.

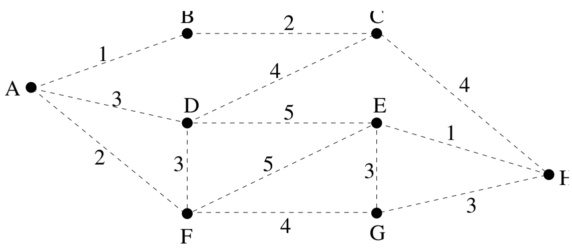


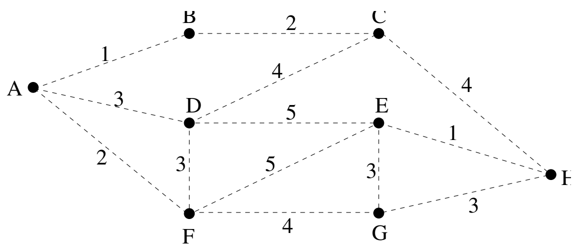
3. Que calculent ces algorithmes ?
4. Est-ce-que l'un des algorithmes est meilleur que l'autre ?

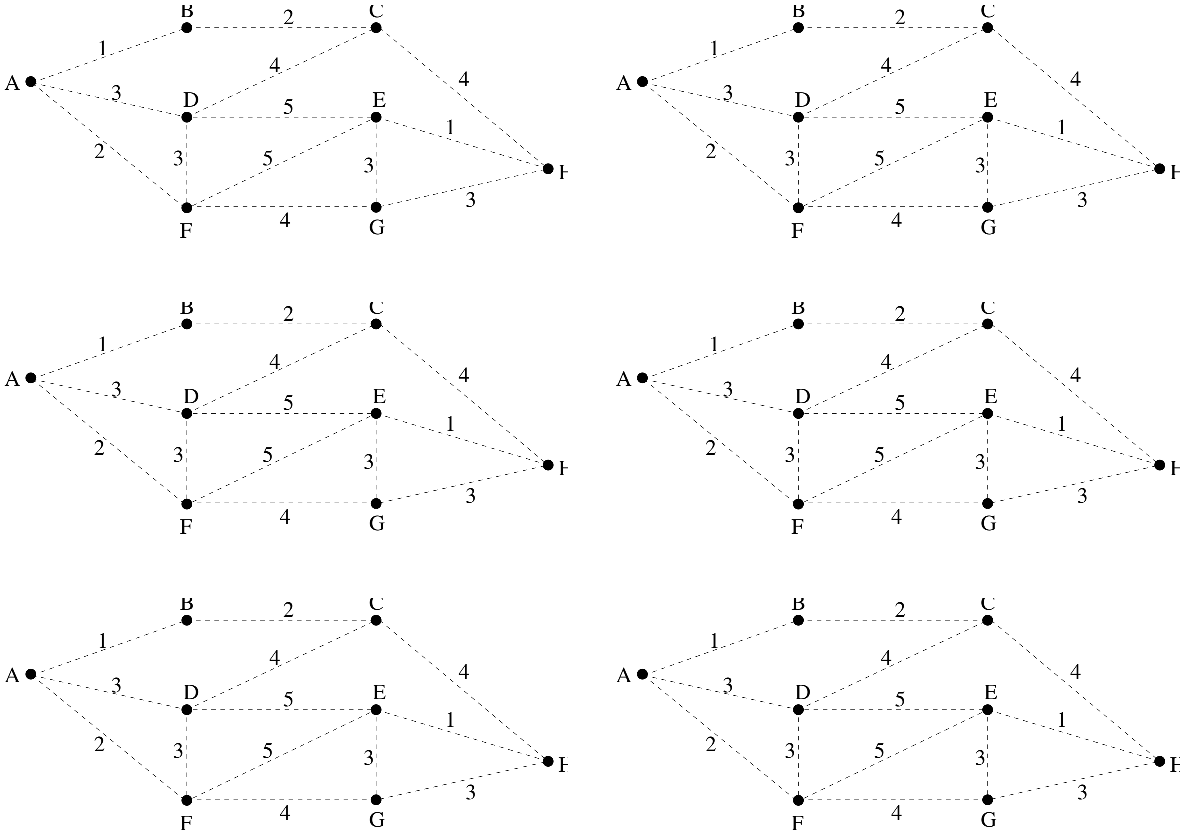
Exercice 3. Appliquer les algorithmes de Kruskal et de Prim au graphe ci-dessous, pour trouver un arbre couvrant de poids minimum (pour l'algorithme de Prim, on commencera par le sommet F).



Algorithme de Kruskal

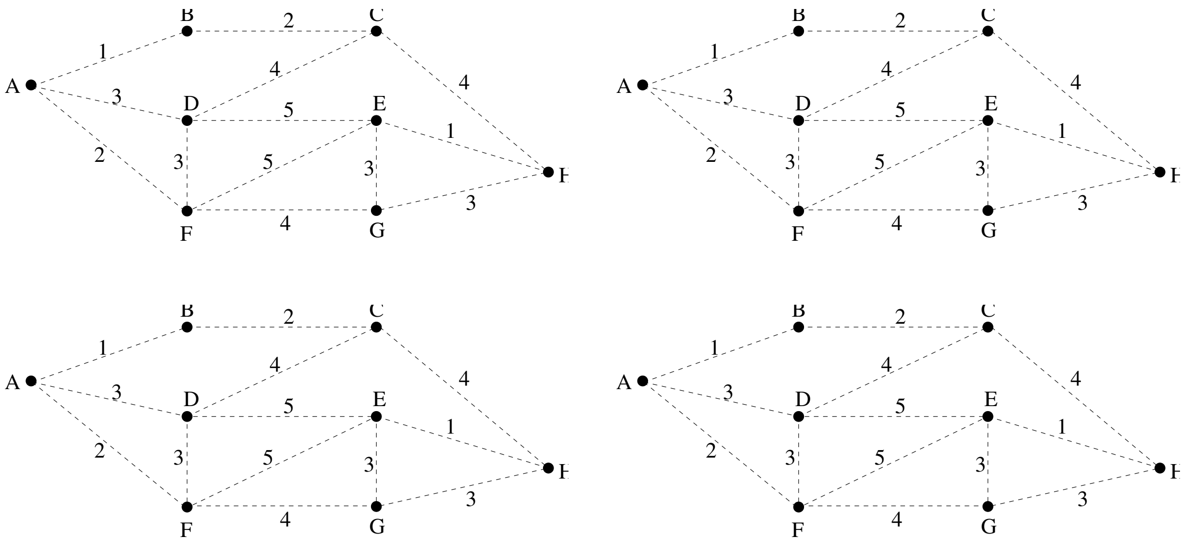


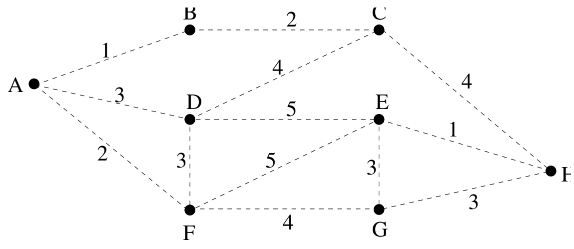


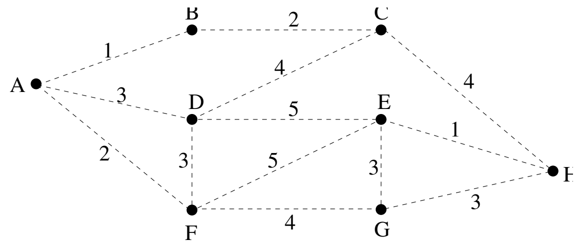


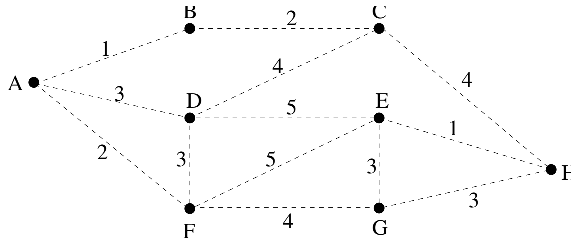
Poids de l'arbre obtenu :

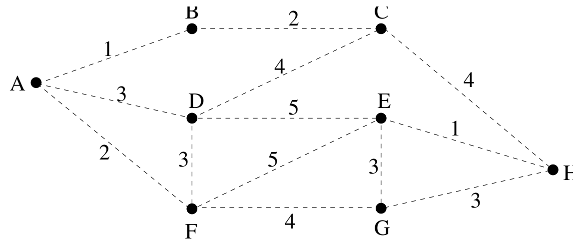
Algorithme de Prim

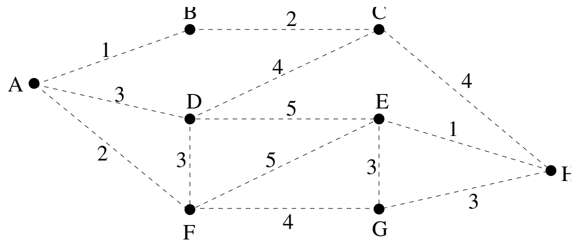


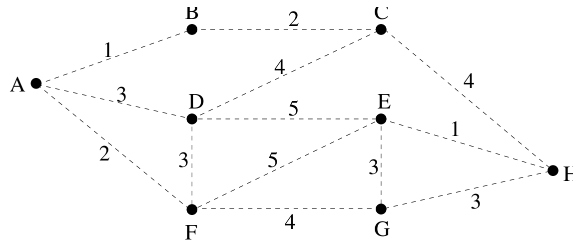






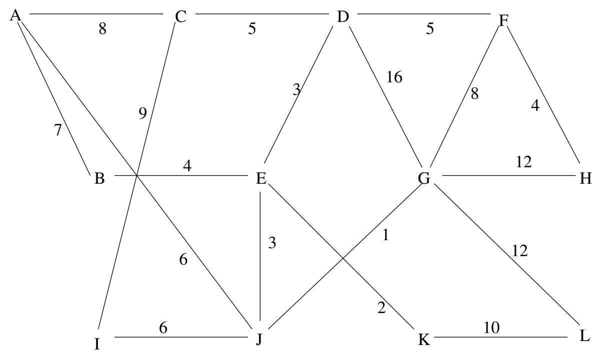




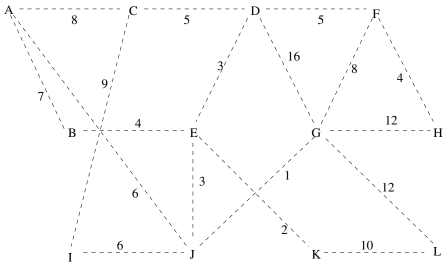


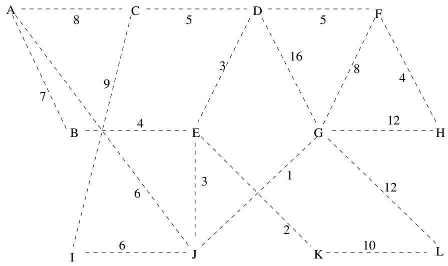
Poids de l'arbre obtenu :

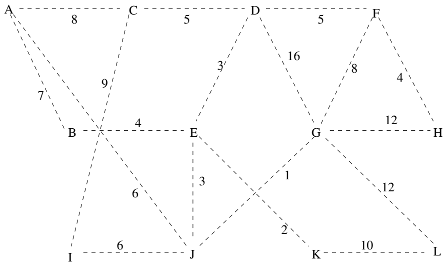
Exercice 4. On considère le graphe valué G ci-dessous :

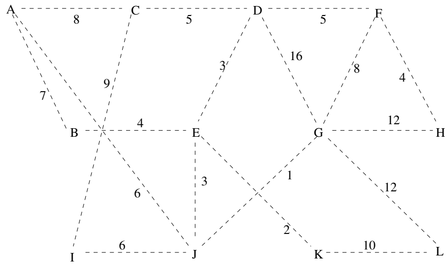


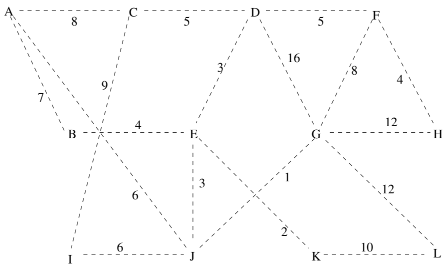
1. Déterminez un arbre couvrant de poids minimal avec l'algorithme de Prim à partir du sommet A. Donnez également le poids de cet arbre. Vous rédigerez votre réponse en détaillant **toutes** les étapes sur le canevas ci-dessous. Vous donnerez également le poids de cet arbre.

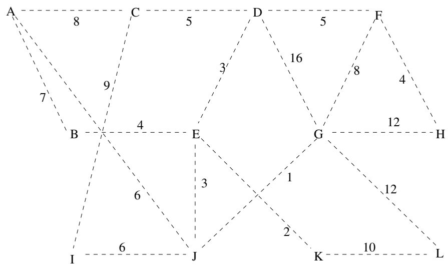

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

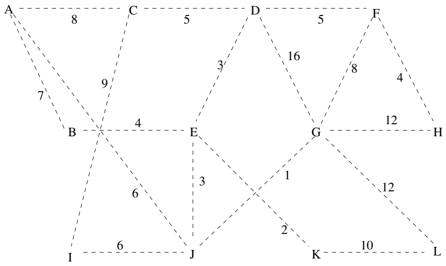

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

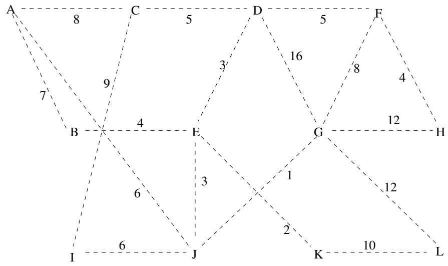

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

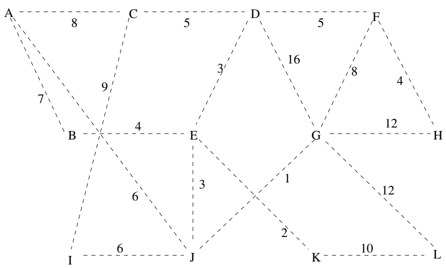

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

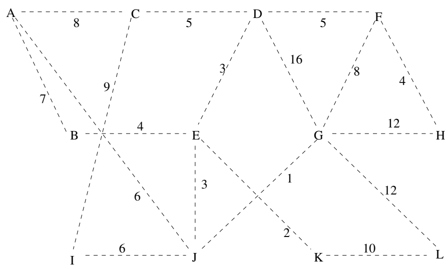

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

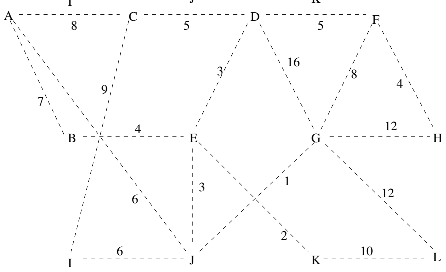

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

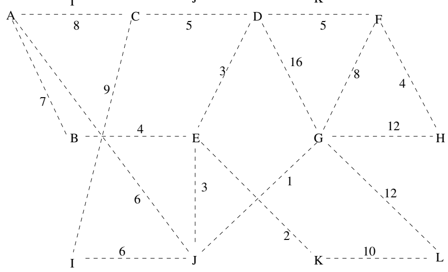

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L


A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L


A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

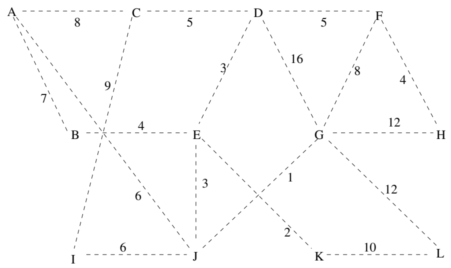

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

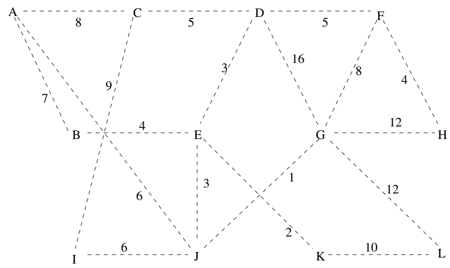

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

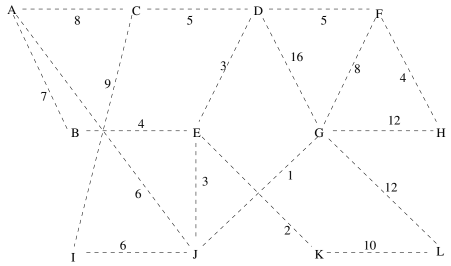

A 8 C 5 D 5 F
7 9 3 16 8 4
B 4 E G 12 H
6 3 1 12 2
I 6 J K 10 L

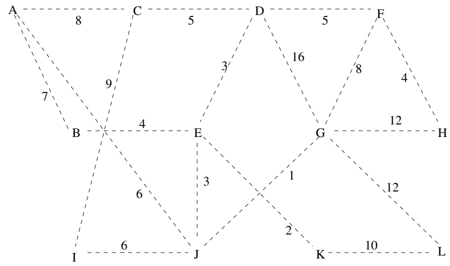
Poids de l'arbre obtenu :

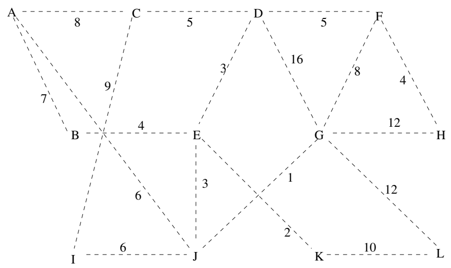
2. Déterminez un arbre couvrant de poids minimal avec l'algorithme de Kruskal. Donnez également le poids de cet arbre. Vous rédigerez votre réponse en détaillant **toutes** les étapes sur le canevas ci-dessous. Vous donnerez également le poids de cet arbre.

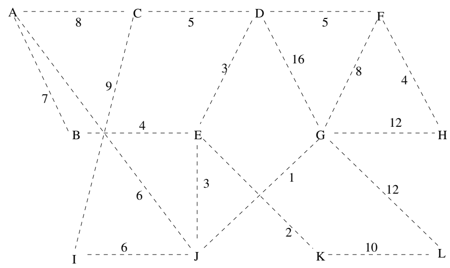


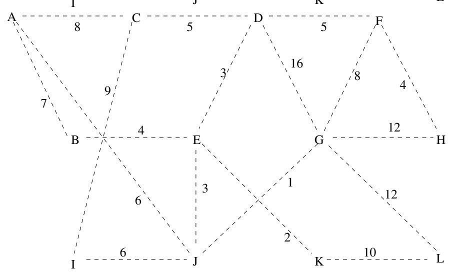


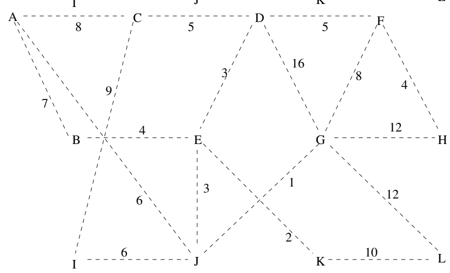


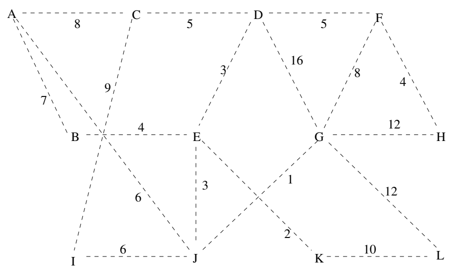


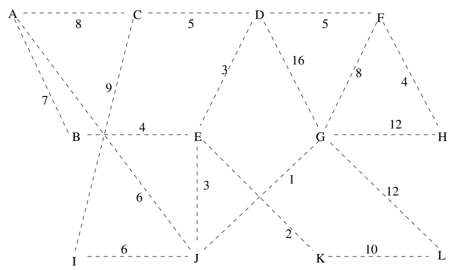


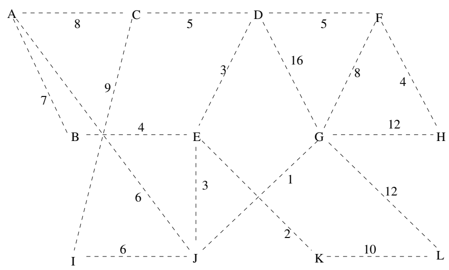


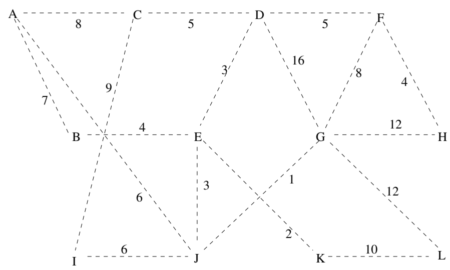












Poids de l'arbre obtenu :

3 Autres exercices

Exercice 5.

1. Trouver tous les arbres à 4 sommets.
2. Trouver tous les arbres à 5 sommets.
3. Trouver tous les arbres à 6 sommets.

Exercice 6.

1. Compter le nombre d'arêtes des arbres à $4, 5$ et 6 sommets construits à l'exercice précédent.
2. Déterminer (avec une preuve) le nombre d'arêtes d'un arbre à n sommets.

Exercice 7. Montrer qu'un graphe connexe a forcément un arbre couvrant.

Exercice 8. Pourquoi l'algorithme de Kruskal construit-il un arbre couvrant de poids minimal ?

Exercice 9. Écrire un programme pour l'algorithme de Prim dans le langage de votre choix. Pour représenter un graphe, on utilisera comme structure de données un tableau à double entrée. Les sommets du graphes seront numérotés de 0 à $n-1$. On conviendra que la case d'indice $[i][j]$ du tableau contient 0 s'il n'y a pas d'arête entre les sommets i et j du graphe, et contient le poids réel $w > 0$ de l'arête entre les sommets i et j du graphe si une telle arête existe. Pour garder la trace des sommets marqués, on utilisera un tableau à n éléments. La case d'indice $[i]$ de ce tableau contiendra 0 si le sommet i n'est pas marqué et 1 sinon.