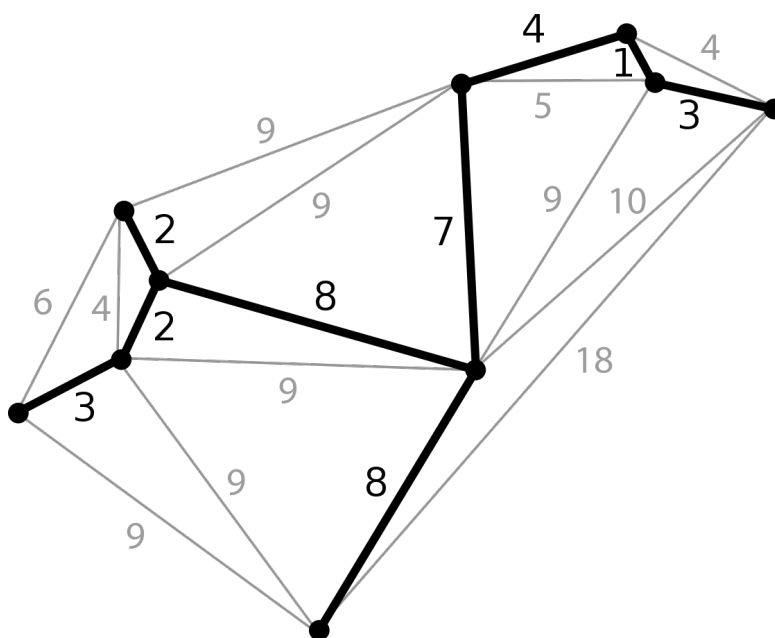


# ARBRE COUVRANT DE POIDS MINIMAL

LA FILIÈRE SYSTÈMES UBIQUITAIRES ET DISTRIBUÉS CLOUD ET IoT



*Equipe :*

ESSAFI Ahmed Yasser

ZEMRANI Nouçair

MOUSAFIA Mohamed El-Mehdi

*Encadrant :*

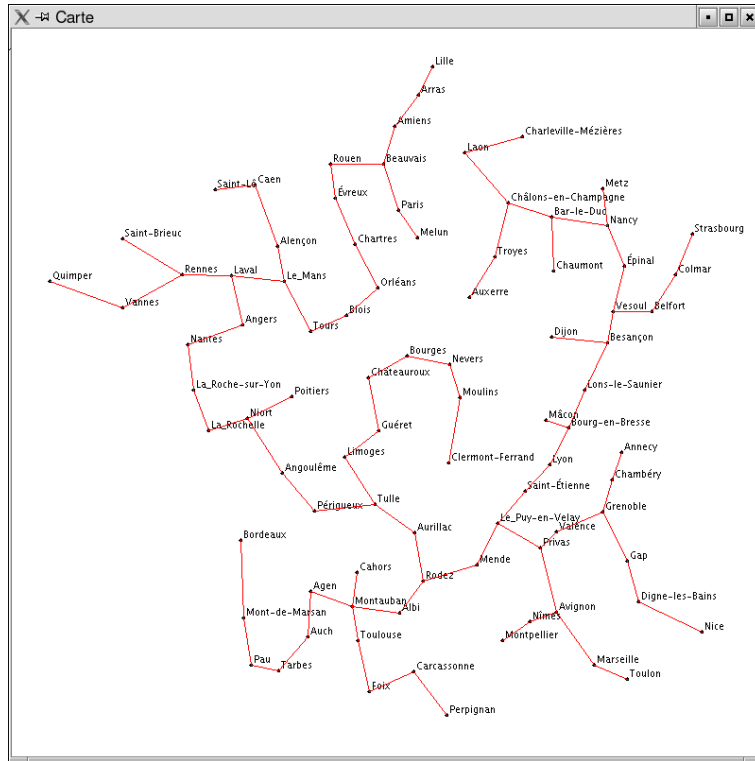
Professeur. BENSAID Hicham

Promotion 2019 - 2022

# Table des matières

Introduction . . . . .	2
Aspects algorithmiques : . . . . .	3
Comparaison : . . . . .	9
Bibliographie : . . . . .	10

# Introduction



Un arbre couvrant de poids minimal (ACM) de ce graphe est un arbre couvrant dont la somme des poids des arêtes est minimale (c'est-à-dire de poids inférieur ou égal à celui de tous les autres arbres couvrants du graphe). L'arbre couvrant de poids minimal est aussi connu sous certains autres noms, tel qu'arbre couvrant minimum.

L'arbre couvrant minimum peut s'interpréter de différentes manières selon ce que représente le graphe. De manière générale si on considère un réseau où un ensemble d'objets doivent être reliés entre eux (par exemple un réseau électrique des habitations), l'arbre couvrant minimum est la façon de construire un tel réseau en minimisant un coût représenté par le poids des arêtes (par exemple la longueur totale de câble utilisée pour construire un réseau électrique).

## Aspects algorithmiques :

### Solution naïve :

On commence tout d'abords par une solution naïve de ce problème.

L'algorithme utilisé crée une liste pour chaque sommets  $L[i]$ , cette liste prend comme valeurs les poids des bords allant du sommets  $i$  aux autres sommets. Pour les sommets ou il n'y a pas un bord qui les relie, on donne la valeur 0. L'utilisateur doit aussi donner la même valeur de poids pour le bord allant de  $i$  à  $j$  ou l'inverse (Même principe pour les autres algorithmes).

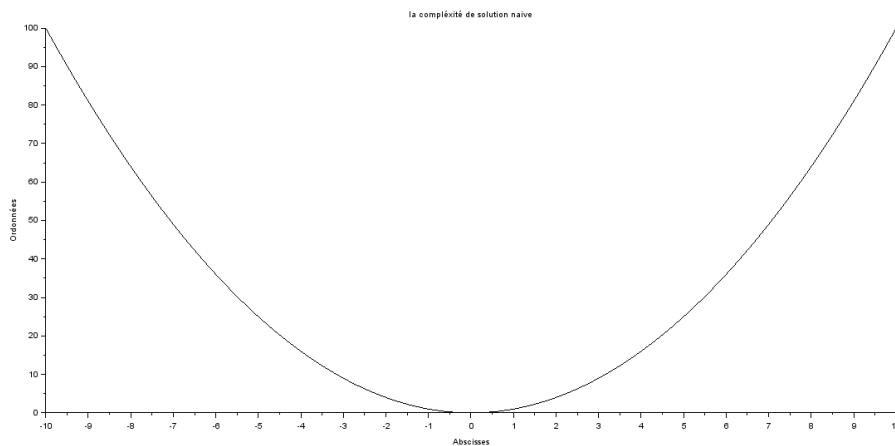
Malheureusement on n'a pas réussi à implémenter une solution optimale, on trouve toujours un bord de plus dans l'arbre couvrant qui est dû au cycle crée.

### **Remarque :**

Les sommets sont : 0,1,2...

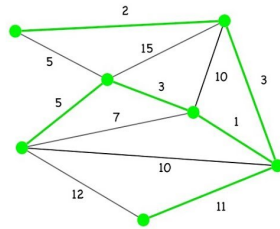
### **complexité :**

la complexité de la solution naïve égale  $O(n^2)$



## Algorithme de Prim :

### Arbre couvrant de poids minimal



L'algorithme de Prim est un algorithme glouton qui calcule un arbre couvrant minimal dans un graphe connexe valué et non orienté. En d'autres termes, cet algorithme trouve un sous-ensemble d'arêtes formant un arbre sur l'ensemble des sommets du graphe initial, et tel que la somme des poids de ces arêtes soit minimale. Si le graphe n'est pas connexe, alors l'algorithme détermine un arbre couvrant minimal d'une composante connexe du graphe.

### *Historique :*

L'algorithme a été développé en 1930 par le mathématicien tchèque Vojtech Jarník, puis a été redécouvert et republié par Robert C. Prim et Edsger W. Dijkstra en 1959. Ainsi, il est parfois appelé DJP algorithm<sup>3</sup>, Jarník's algorithm, Prim–Jarník algorithm, ou Prim–Dijkstra algorithm.

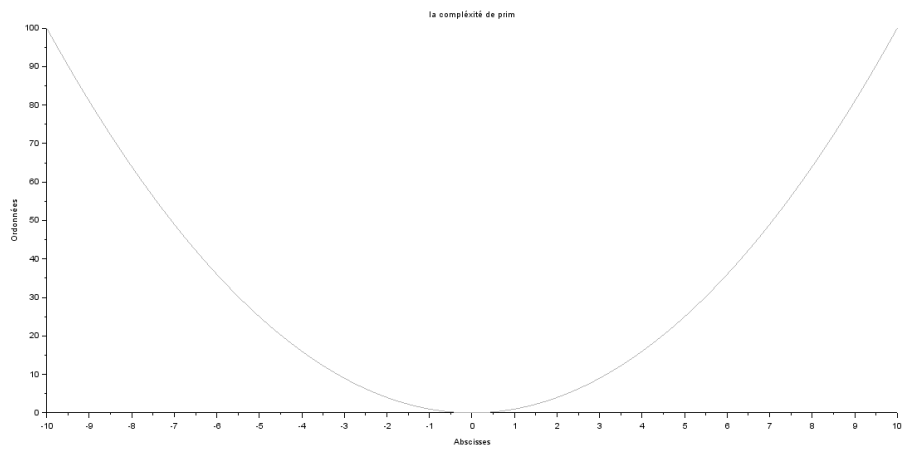
### *Principe :*

1. Dans un premier temps tous les domaines se produire Touche  $[v]$  à  $+$  et tous les champs  $[v]$  à NIL.
2. Il faut tout sommet comme la racine et met sa clé à 0.
3. Ils intègrent tous les sommets étaient dans une structure de données appropriée (généralement un file d'attente prioritaire) Et les extraits dans l'ordre croissant.
4. Elle parcourt ensuite la liste des contiguïtés du sommet extrait ( $u$ ) En ne considérant que les sommets ( $v$ ) Toujours dans la structure auxiliaire.
5. Pour chacun d'entre eux de telle sorte que sa distance par rapport  $u$  est la plus courte parmi tous ceux qui sont considérés, se pose  $[v]$  égal à  $u$  l'insertion, en fait,  $v$  MST.
6. Ceci conclut le cycle de mise à jour du champ Touche  $[v]$  avec la valeur de la distance entre  $u$  et  $v$ .

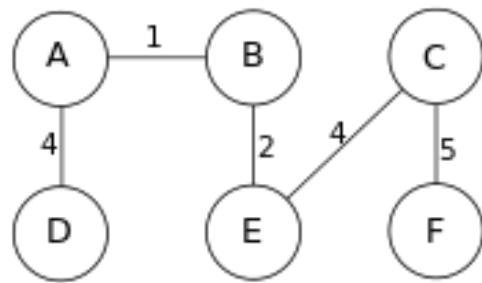
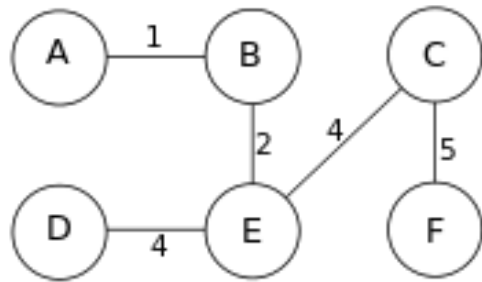
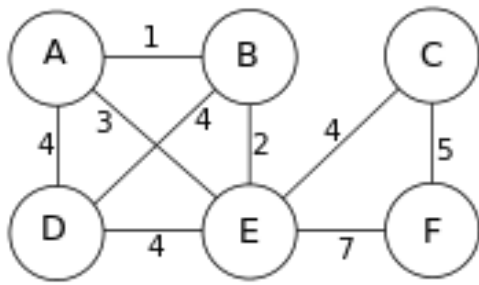
### *Complexité :*

la complexité de Prim est égale à  $(O(n^2))$

( Même valeur que la complexité de l'algorithme trouvé dans les sites Internet )



### Algorithme de Kruskal :



L'algorithme de Kruskal est un algorithme minimum spanning-tree qui trouve un bord du moins de poids possible qui relie les deux arbres dans la forêt. Il est un algorithme glouton en théorie des graphes qu'il trouve un arbre couvrant minimal pour un connecté graphe pondéré en ajoutant l'augmentation des arcs de coûts à chaque étape.

### **Remarque :**

Malheureusement l'algorithme crée n'est pas optimale et on risque toujours de trouver un bords de moins ou de plus dans l'arbre couvrant de poids minimal ( le cas où il y'a plusieurs cycles dans l'arbre )

### **Historique :**

l'algorithme de Kruskal est un algorithme de recherche d'arbre recouvrant de poids minimum

(ARPM) ou arbre couvrant minimum (ACM) dans un graphe connexe non-orienté et pondéré. Il a été conçu en 1956 par Joseph Kruskal.

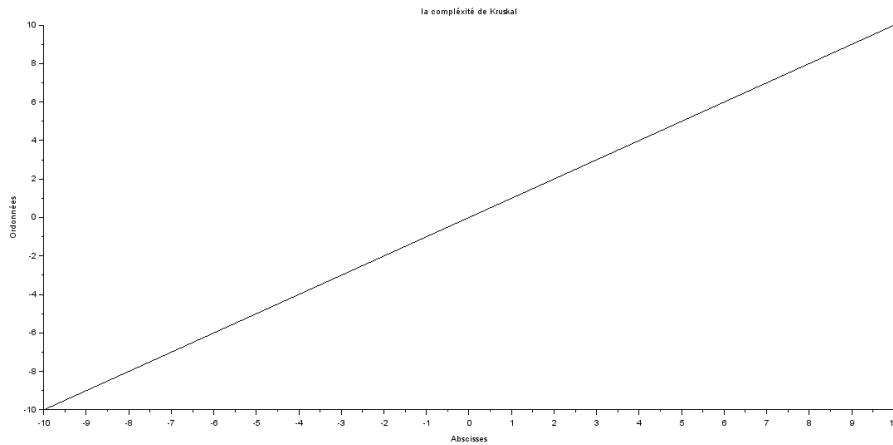
### ***Principe :***

1. créer une forêt F (un ensemble d'arbres), chaque sommet du graphe est séparée arbre.
2. créer un ensemble S qui contient toutes les arêtes dans le graphe.
3. tandis que S est non vide et F est pas encore Spanning.
  - retirer un bord avec un poids minimum de S
  - si le bord enlevé relie deux arbres différents ajouter ensuite à la forêt F, la combinaison de deux arbres dans un seul arbre

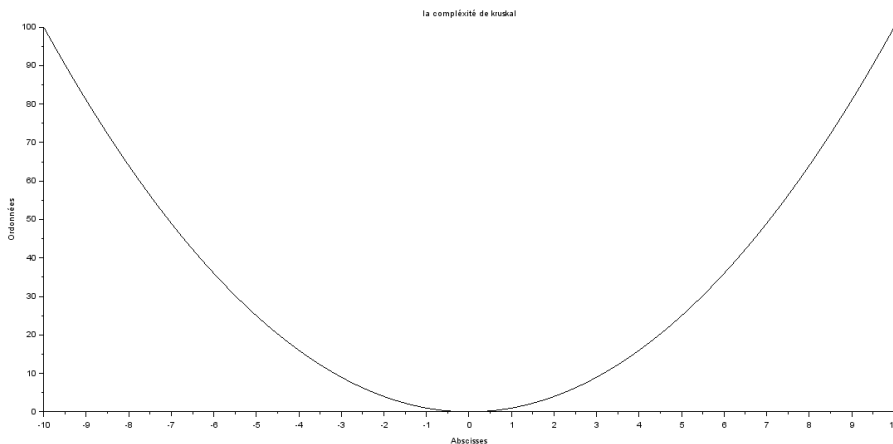
A la fin de l'algorithme, la forêt forme une forêt couvrant minimal du graphique. Si le graphique est connecté, la forêt a un seul composant et forme un arbre couvrant minimal.

### ***complexité :***

la complexité au meilleure cas :  $O(n)$



la complexité au pire cas :  $O(n^2)$





***Remarque :***

la complexité de l'algorithme est de coût linéaire par rapport à  $n$  si on ne l'optimise pas, mais peut être de coût  $\log(n)$  si on utilise une structure de données de type anti-arborescence.

La complexité de l'algorithme est ainsi celle d'un tri de  $m$  arêtes.

## Comparaison :

### Comparaison entre la solution naive et l'algorithme de Prim et celui de Kruskal

On remarque que la complexité de la solution naive ressemble a celle de l'algorithme de Prim et de kruskal au cas linéaire (  $O(n^2)$  ).

La Solution naive ne donne pas un résultat optimal contrairement aux autres algorithmes.

### Comparaison entre l'algorithme de Prim et de Kruskal

- L'algorithme de Prim s'initialise avec un nœud, alors que l'algorithme de Kruskal commence avec un bord.
- Les algorithmes de Prim varient d'un noeud à l'autre tandis que l'algorithme de Kruskal sélectionne les arêtes de façon à ce que la position du bord ne soit pas basée sur la dernière étape.
- Dans l'algorithme de prim, le graphe doit être un graphe connecté alors que le Kruskal peut aussi fonctionner sur des graphes déconnectés.
- la complexité

## Bibliographie :

- Algorithme de Prim [[https://fr.wikipedia.org/wiki/Algorithme\\_de\\_Prim](https://fr.wikipedia.org/wiki/Algorithme_de_Prim)]  
<https://boowiki.info/art/les-algorithmes-d-optimisation/algorithme-de-prim.html>
- arbre couvrant de poids minimal [[https://fr.wikipedia.org/wiki/Arbre\\_couvrant\\_de\\_poids\\_minimal](https://fr.wikipedia.org/wiki/Arbre_couvrant_de_poids_minimal)]
- (en)[PDF] James Allen Fill et J. Michael Steele, Exact expectations of minimal spanning trees for graphs with random edge weights [archive], 2004
- Algorithme de Kruskal [<https://fr.qwe.wiki/wiki/Kruskal>]