Cours Algorithmique.

Les arbres de recouvrement de poids minimum.
(ARPM / ARM)

~~Nous Soit graves~~

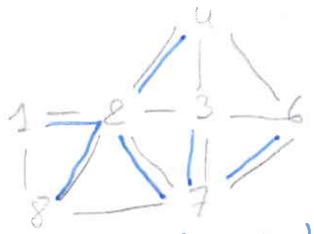
Soit des graphes valués non orienté :

__Définition__ : Un arbres est un graphe connexe est sans cycle (non-orienté)

. Un arbre de recouvrement : c'est un graphe partiel qui conserve les sommets en étant connexes et sans cycle.

On cherche un arbre de recouvrement de poids minimum, c'est-à-dire que le cout de la somme de toute les arrêtes sont minimum.

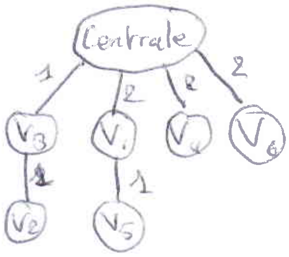
Les deux algos classiques pour ce calcul sont : prim et Kruskal (graphes non orienté)
et edmonds (graphe orienté)



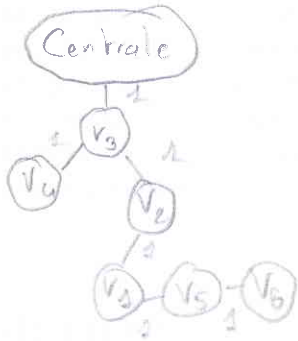
- arrête de l'arbre de recouvrem-

Soit $G = \langle S, A \rangle$ un graphes non-orienté de n sommets, c'est un arbre ssi il est __connexe__ et __sans cycles__.

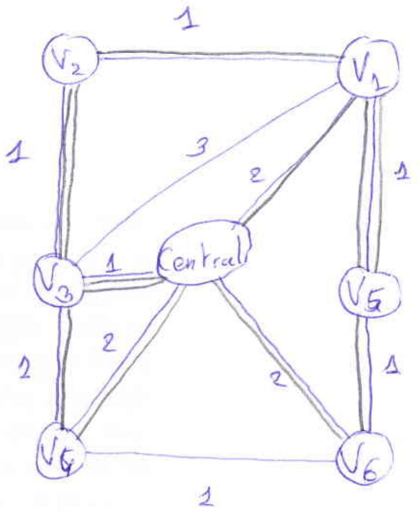
↳ "si on supprime une arrête qqcq, il n'est plus connexe".
↳ "il est est connexe avec (n-1) arrête".
↳ "il est sans cycle si on ajoute une arrête on en crée un".
↳ "il est sans cycle avec (n-1) arrêts".
↳ "Tout coupe de sommet S est reliée par une unique chaine".



Arbre de plus court chemin.



Arbre de recouvrem-



- graphe d'origine
- arbre de recouvrement
- algo de plus court chemin.

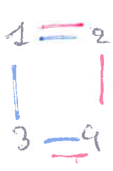
Conditions d'existance

En cas de graphes non-connexe on applique l'algo pour obtenir une forêt de recouvrem- (nbre d'arbre de recouvrem- > 1).

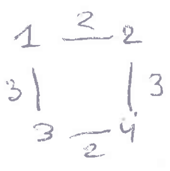
__Remarque__ : Rechercher un arbre de recouvrem- non connexe, ça revient à rechercher un graphe partiel de poids minimum. Ce dernier ne possède pas de cycle car plus lourd.

__Condition d'unicité__ : Dans le cas d'un graphe ~~non~~ connexe, peut-on avoir ~~un~~ plusieurs arbre de recouvrem- minimum ?

Soit un graphe G non-o valué, tq les coups sont tous distincts de A2. Alors G admet un unique arbre de recouvrem- min.
Ɪ le cout de deux arcs n'est pas identique.



T_1 T_2



G

Arbre non orienté connexe valué : L prim.

* Il conserve la connexité avec la source, proche de Djikstra.

KRUSKAN : Conserve l'acyclicité

EDMONDS. (Graphe orienté valué) : Input : Sommet de départ.
Output : Un arbre qui à pour racine l'Input.