**Chapitire 7 Graphes valué : Arbre couvrant de poids minimum**

valutation (fonction de poids/fonction de coût) :

p : E -> R

les arêtes ou les arcs qui ont des poids et non les sommets

graphe valué : graphe muni d’une valuation

poids/coûts totals : G = (X,E)

- pour G : somme des poids des arêtes, p(G) = sigma p(l)

- graphe partiel de G’ = (X,E’) : sommets des poids des arêtes, p(G’) = sigma p(l)

- chemin C = (X1,X2….Xn) : p(C) = sigma p((Xi,X(i+1))

représentation d’un graphe valué :

- par liste d’arêtes : arête (i, j) de poids p(i, j) représenté par un triplet (i, j, p(i, j))

- par matrice de poids : p(i,j) si l’arc (i, j) appartient à G, valeur conventionnelle ex : 0, infini si arc n’existe pas

- par liste d’adjacence : associé à sa matrice de poids

arc couvrant : graphe partiel qui soit un arbre

- un graphe G = (X,E) admet un arbre couvrant ssi il est connexe

Algorithme Kruskal :

- Trier les arêtes par poids croissant et les stocker dans une liste L

Initialiser liste T à [ ], T va contenir des arêtes de l’arbre couvrant cherché

À chaque étape on ajoute à T l’arête de poids minimal dont l’ajout n’engendre pas de cycle (T n’est pas connexe au cours de la construction)

Si on parvient à la m = n-1ième étape à un graphe n sommets et m = n-1 arêtes on a un graphe sans cycle de n-1 arêtes, i.e. un arbre. Comme il est élémentaire, il passe par n sommets et est donc couvrant. Tous sous-graphe de poids strictement inférieur contient des cycles et n’est donc pas un arbre. Finalement T obtenu est bien un arbre couvrant de poids minimal. Si avant la mièrme étape, il n’y a plus d’arête qui convient c’est que le graphe initial n’était pas connexe.

Complexité :

- à la ième étape, T possède donc un parcours en largeur de complexité O(i)

- il y a m-i arêtes à tester

- le pire cas : la dernière la bonne ce qui donne une étape i de coût (m-i)\*i, donc sigma (m-i) \* i = O(m^2) (O(n^4) pour un graphe complet)