Arbre

H: un graphe non orienté et ayant n sommets

Définitions 1 (d'un arbre):

- H est connexe et sans cycle
- H est connexe et comporte n-1 arêtes
- H est sans cycle et comporte n-1 arêtes
- H est sans cycles et l'ajout d'un arête crée un cycle unique
- H est connexe et la suppression d'une arête le rend non connexe
- 2 sommets quelconques sont reliés par une chaîne unique

Arbre recouvrant minimal

On suppose que G =(X,U) est un graphe connexe, et qu'à chaque arête u de G est attribué un coût l(u).

Définition 2 (Arbre recouvrant)
H (X,V) est un arbre recouvrant de G si V est un sous ensemble de U.

Définition 3 (Coût d'un arbre recouvrant)
Le coût de H est défini par $C(H) = \sum_{v \in V} l(v)$

Définition 4 (Arbre recouvrant minimal)
H sera dit arbre recouvrant minimal, si son coût est le minimum des coûts de tous ses arbres recouvrants.

Problème déterminer un arbre recouvrant minimal d'un graphe connexe et valué G

Algorithme de Prim

G=(X,U) un graphe connexe :

Procédure PRIM (G : graphe, var H : ensemble d'arêtes)

Y : ensemble de sommets;

x , y : sommets;

Début

H=Φ ; Choisir un sommets de X soit x_i (ce sommet); Faire Y:={x_i};

Tant que Y # X

Début

Soit [x,y] une arête de coût minimal tel que $x \in Y$ et $y \in X-Y$;

H := $H \cup \{(x,y)\}$;

Y:= $Y \cup \{y\}$

FinTanque

Fin;

Algorithme de KRUSKAL

G=(X,U) un graphe connexe

Procédure KRUSKAL (G : graphe, var H : ensemble d'arêtes)

L : Liste d'arêtes

K : entier :=0;

Début

Etablir une liste L des arêtes classées par ordre croissant de leur coût;

Faire H := Φ ;

Tant que k<n-1

Debut

Soit u la première arête de L non déjà traitée;

Si (u ne crée pas un cycle avec les arcs déjà retenu dans H) Alors

H := H$\cup${u}; K:=K+1; (marquer u : visitée) Finsi

FinTanque

Fin (Kruskal)

Algorithme de Sollin

G=(X,U) un graphe connexe

Procédure SOLLIN (G : graphe, var H : ensemble d'arêtes)

x,y : sommets; T : graphe;

k : entier :=0;

Début

H=Φ;T:=(X,H); (Le graphe T contient n composantes connexes formées par les singletons {x})

Tant que K< n-1

Début

Choisir une composante connexe *C* de T;

Choisir une arête [x,y] de coût minimal parmi toutes les arêtes ayant une extrémité dans *C* et l'autre extrémité dans X-*C*.

H:=H ∪{[x,y] } ; T=(X,H);

k:=k-1;

Fin tant que

Fin (SOLLIN)

∪