E.D. Algorithmes et Structures de Données n° 6

Thème : Algorithme de Kruskal
(recherche d'un arbre couvrant de poids minimal dans un graphe connexe)

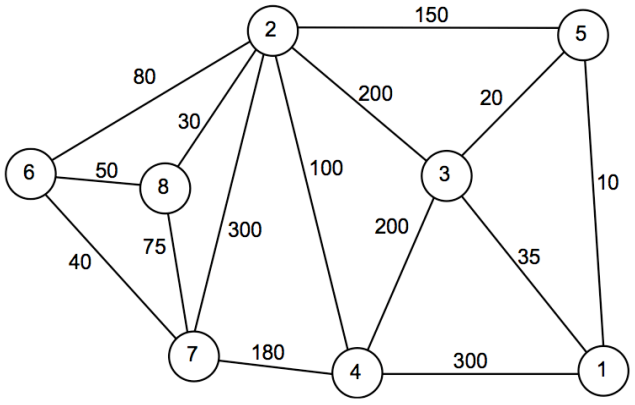
Exercice VI.1

Question 1 L'algorithme de Kruskal

Rappel: un graphe G de n sommets et m arêtes est un arbre ssi il est connexe et m = n-1.
Rappel de l'algorithme:

```
Entrée : G = (X,U,P) n sommets, m arêtes
Sortie : A arbre couvrant de poids minimal de G
procedure kruskal (G : in graphe; A : out arbre) :
      k : entier := 0; -- Nombre d'arêtes choisies
      V : {arête} := ∅
début
      trier les arêtes de G par ordre de poids croissant;
      tant que k < n-1 faire
            parcourir la liste triée des arêtes;
            sélectionner la première arête w qui ne forme pas de
            cycle avec les précédentes;
            k := k+1;
            V := V ∪ {w};
      fait
      A := graphe(X,V);
fin;
```

Appliquer l'algorithme à l'exemple suivant :



Question 2 Le tri des arêtes

`Soit le tableau suivant donnant les arêtes du graphe et leur poids dans un ordre quelconque :`

150	200	100	300	80	30	50	40	75	180	20	200	35	10	300
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Appliquer l'algorithme de tri par tas pour trier les arêtes par ordre de poids croissants.

Question 3 Détection des cycles

3.1. Le principe

Comment peut-on détecter qu'une arête sélectionnée forme un cycle avec les précédentes ? Au départ, aucune arête n'a été choisie. On considère que les sommets forment chacun un ensemble connexe réduit à un singleton. Puis, à chaque fois qu'on choisit une arête, il faut que les deux extrémités de cette arête soient dans deux ensembles différents, sinon on formerait un cycle. Enfin, le fait d'ajouter une arête implique qu'on fusionne deux ensembles en un même ensemble connexe.

3.2. L'algorithme

Il faudra répondre aux questions suivantes :

- Comment tester si 2 sommets x et y sont dans le même ensemble ?
- Comment fusionner 2 ensembles associés à 2 sommets x et y ?

On peut utiliser les variables suivantes, dans lesquelles "cc" signifie "composante connexe"

- un tableau qui donne, pour un sommet donné, l'ensemble auquel il appartient. Cet ensemble sera représenté par une liste de numéros de sommets. Ce tableau sera initialisé par des listes contenant un sommet chacune. On appellera ce tableau `tabEnsSommets,` il sera de type `Liste[]` où Liste représente une liste d'entiers et il aura autant d'éléments que de sommets dans le graphe.
- au moment de la fusion des ensembles associés à deux sommets x et y, on ajoutera la liste de x à celle de y (et la liste de x devient vide). Il faut alors indiquer que pour trouver la liste de x, il faut aller à y. Pour ce faire, on utilisera un tableau qui pour chaque sommet x, donne le numéro du sommet suivant qu'il faut regarder pour trouver la liste où apparaît x. On appellera ce tableau `tabMemeEns,` il sera de type `int[]` et il aura autant d'éléments que de sommets dans le graphe. On initialisera tous les éléments à -1 pour indiquer qu'au départ il n'y a pas de redirection.