DIOUBATE SEKOU

Traitement des graphes

Projet en théorie des graphes

**La théorie des graphes** est très probablement née en 1735 lorsque Leonhard Euler (1707 - 1783) résout le problème des sept ponts de Königsberg (De nos jours Kaliningrad en Russie). Ce problème est très simple sur le principe mais un peu plus compliqué à démontrer, en voici l'énoncé : La vile de Königsberg est une ville autour d'un fleuve, elle compte quatre berges et sept ponts les reliant. Le but du jeu est de savoir s'il existe un chemin permettant d'emprunter tous les ponts une fois et une seule et revenir au point de départ. Le problème s'appelle, de façon plus formelle, la recherche d'un cycle eulérien dans un graphe. Euler a démontré que ce problème n'avait pas de solution.

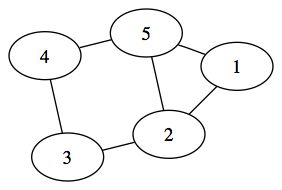
**Définition**

En informatique, **un graphe** c’est une structure de donnée tout comme les tableaux, composée de nœuds interconnectés entre eux par des liens. Chaque nœud contient une ou plusieurs données de n’importe quel type.

**Un graphe** est composé de **sommets** et d'**arcs** (ou d'**arêtes**). Un graphe G est défini de manière formelle par un couple (S, A) où :

* **S** est un ensemble fini d'éléments. Chacun de ces éléments est appelé sommet du graphe.
* **A** est un sous ensemble (éventuellement nul) de SxS. Chacun de ces éléments de A est appelé arc ou arête.

L'ensemble A est donc composé de paires (x,y), x et y étant appelés les extrémités de l'arête (ou de l'arc). Cette définition peut paraître très vague, voici donc un exemple de graphe :



*Exemple de graphe*

public class sommet

{

public int numero;

public int x, y;

public int cout;

public sommet precedant;

public bool marquer = false;

public List<liaison> Liaisons\_sortants;

public List<liaison> Liaisons\_entrants;

public sommet()

{

x = 0; y = 0; numero = 0; cout = -1;

Liaisons\_sortants = new List<liaison>();

Liaisons\_entrants = new List<liaison>();

}

public sommet(int \_numero, int \_x, int \_y, int \_cout)

{

x = \_x; y = \_y; numero = \_numero; cout = \_cout;

Liaisons\_sortants = new List<liaison>();

Liaisons\_entrants = new List<liaison>();

}

}

public class liaison

{

public sommet sommet\_lier;

public int cout;

public liaison()

{

sommet\_lier = null; cout = -1;

}

public liaison(ref sommet \_sommet\_suivant, int \_cout)

{

sommet\_lier = \_sommet\_suivant; cout =\_cout;

}

}