

TP N°02 : Rappel sur les concepts de base de la théorie des graphes

Objective TP: Créer et afficher un graphe de différentes manières.

Remarque : Ecrivez des commentaires dans vos programmes.

Exercice 2.1 (Graphes à prendre en compte)

Ecrire un programme qui doit être capable de prendre en compte n'importe quel graphe orienté valué, sachant que :

- Les sommets sont des numéros entiers de '0' à 'n-1' pour un graphe contenant n sommets ;
- Les valeurs associées aux graphes sont des nombres entiers quelconques ;
- Il y a au plus un arc reliant deux sommets ;
- Il n'y a pas de boucle ;
- Il peut y avoir des sommets isolés (sans prédécesseur ni successeur).

Exercice 2.2 (Création d'un graphe aléatoire G à n sommets et m arêtes (méthode 1))

L'ensemble des sommets de G est codé par $\{0, \dots, n-1\}$. L'ensemble des m arêtes de G est stocké dans un tableau **edge** de taille $m \times 2$ et dont les entrées appartiennent à $\{0, \dots, n-1\}$. Ainsi, si xy est l'arête de G d'indice k, on aura **edge[k][0]=x** et **edge[k][1]= y**.

Par exemple, le graphe sur l'ensemble de sommets $\{0, 1, 2, 3\}$ ayant pour arêtes 01, 02, 03, 12, 23 est codé par le tableau **edge[5][2]={0,1},{0,2},{0,3}, {1,2},{2,3}**.

Question : Ecrire une fonction **void grapherandom(int n, int m, int edge[][2])** qui engendre aléatoirement le tableau **edge** en tirant au hasard chacune de ses entrées. On permettra la création d'arêtes multiples ou de boucles.

On pourra utiliser les appels :

- **srand (time(NULL))** // Initialise la graine (seed) de la fonction rand sur l'horloge.
- **rand()%k** // Retourne un entier entre 0 et k - 1.

Exercice 2.3 (Création d'un graphe aléatoire(méthode 2))

Ecrire une fonction **void voisinstrandom(int n, int m, vector<int> voisins[])** qui engendre aléatoirement les listes de voisins d'un graphe aléatoire de n sommets et m arêtes. On prendra garde a :

- la symétrie: si x est voisin de y, alors y est voisin de x.
- ne pas créer de boucle.
- ne pas créer d'arête multiple.

Exercice 2.4 (Affichage du graphe)

Ecrire **void AffichageGraphe(int n, int m, int edge[][2])**, qui permet d'afficher le graphe créé dans l'exercice 2.1, 2.2 et 2.3.