Initiation à l'informatique - 1er semestre

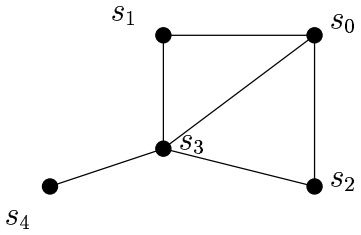
TP 2

Type abstrait centré-sommets

On considère le graphe en tant que type abstrait comme un ensemble de sommets et un ensemble d'opérateurs sur ces sommets.
On appelera ce type abstrait: *type abstrait centré-sommets.*

Exercice 2.1 *Premiers pas*

1. Téléchargez les fichiers suivants :
   - `vgraph.py` qui contient une implémentation python du *type abstrait centré-sommets*;
   - `graph.py` qui regroupe des parties communes des deux types abstraits *centré-sommets* et *centré-arêtes*;
   - `utils.py` qui est composé par des définitions des opérations utiles aux deux types abstraits;
   - `graph1.py` qui contient la definition en tant que graphe de *type abstrait centré sommets* du graphe illustré ci-dessous:



   - `exercice2-1.py` qui contient un programme permettant de réaliser vos premières opérations sur les graphes.

2. Lancez Emacs (en tapant `emacs &`).

3. Ouvrez le fichier `exercice2-1.py` (menu `File > Open File...`).
   Ce fichier contient le programme python suivant:

```
print "Number of nodes of gr1 : ", NB_NODES(gr1)
n0 = NODE(gr1, 0)
print "Number of edges of node 0 : ", NB_EDGES(n0)
n3 = NODE(gr1, 3)
print "Number of edges of node 3 : ", NB_EDGES(n3)
if (EDGE(n0,2) == EDGE(n3,0)):
print "Edge 2 at node n0 and edge 0 at node n3 are the same"
```

4. Démarrez l'interpréteur python (menu `Python > Start interpreter`).

5. Saisissez les lignes suivantes:

```
from vgraph import *
execfile("graph1.py")
```

6. Exécutez le programme (menu `Python > Execute Buffer`). Essayez de comprendre l'effet de chaque commande en étudiant chaque ligne du programme.

Exercice 2.2 *Liste de sommets et leurs degrés respectifs*
Écrivez une fonction `list_of_vertex_degrees` qui prend comme paramètre un graphe G et affiche la liste de ses sommets avec leurs degrés respectifs.
Appliquée au graphe de l'exemple `graph1.py`, la fonction affichera:

```
the vertex <Node 0> is of degree  3
the vertex <Node 1> is of degree  2
the vertex <Node 2> is of degree  2
the vertex <Node 3> is of degree  4
the vertex <Node 4> is of degree  1
```

Exercice 2.3 *Somme des degrés des sommets*
Écrivez une fonction `sum_of_vertex_degrees` qui prend comme paramètre un graphe G et renvoie la somme des degrés de ses sommets.
Sur le graphe de l'exemple `graph1.py`, le résultat sera 12.

Exercice 2.4 *Degré maximum*
Implémentez une fonction `highest_degree` qui prend comme paramètre un graphe G et qui renvoie le degré maximum des sommets du graphe G.
Sur le graphe de l'exemple `graph1.py`, le résultat sera 4.

Exercice 2.5 *Sommets voisins*
Écrivez une fonction `is_neighbor_of` qui prend comme paramètres deux sommets s et t et qui renvoie 1 si s est adjacent à t et 0 sinon.
Pour l'exemple `graph1.py` et les couples de sommets s_0 et s_1, s_1 et s_2, un programme test affichera:

```
s0 and s1 are neighbour :  1
s1 and s2 are neighbour :  0
```

Exercice 2.6 *Nombre de sommets de degré fixé*
Écrivez une fonction `nb_vertices(G,d)` qui prend comme paramètres un graphe G et un entier d, et qui renvoie le nombre de sommets de degré d.
Pour le graphe de l'exemple `graph1.py` et l'entier $d = 2$ un programme test affichera:

```
Number of vertices of degree 2:  2
```