Informatique 2eme année, CPBX, TD5

Paul Dorbec

Ce TD traite des graphes, et est largement inspiré d'un TD préparé par l'équipe d'initinfo.

1 Graphes

Un *graphe* est une modélisation d'un ensemble (non vide) d'objets reliés : les objets sont appelés *sommets*, et les liens entre les objets sont appelés *arêtes*. Deux sommets reliés par une arêtes sont dits *voisins*.

On peut utiliser les graphes pour représenter diverses situations courantes : les liens d'amitiés entre étudiants, les matches entre équipes de sport, les liaisons moléculaires entre des atomes, les routes entres les villes, ...

Nous vous fournissons une définition python de la *classe* des graphes, ainsi que des autres classes (sommets et arêtes) nécessaires. Ces définitions se trouvent dans le module `graphV3.py`. Ce module comporte aussi une vingtaine de fonctions qui permettent de manipuler graphes, sommets et arêtes sans connaître les détails des classes correspondantes. Un autre module `bibV3.py` contient un certain nombre de graphes construits en utilisant `graphV3.py` ; pour utiliser un module il faut commencer par l'*importer* avec

```
from bibV3 import *
```

Le module `bibV3` importe à son tour le module `graphV3` ; les noms de ces modules se terminent par "V3" pour indiquer qu'ils sont adaptés à la Version 3 du langage python.

Pour python, l'objet `tgv2005`, est de type `<graphe: 'tgv2005'>` pour indiquer que cet *objet* est un graphe. On peut obtenir la liste complète des sommets d'un graphe G avec la fonction `listeSommets(G)`, par exemple en tapant `listeSommets(tgv2005)` on obtient la liste :

```
[<sommet: 'Lille', 'white', False>, <sommet: 'Paris', 'white', False>,
 ...
 <sommet: 'Bordeaux', 'white', False>,
 ...
```

```
<sommet: 'Strasbourg', 'white', False>]
```

Chaque sommet est doté d'un nom, d'une couleur et d'une marque (booléenne). Le *nom* d'un sommet, par exemple `'Bordeaux'`, est un simple attribut du sommet sous forme de chaîne de caractères. On peut utiliser la fonction `sommetNom(G,etiquette)` pour accéder au sommet par son étiquette : `sommetNom (tgv2005, 'Bordeaux')` est une expression correcte pour désigner ce sommet.

`listeSommets(G)`	retourne la *liste* des *sommets* de `G`
`nbSommets(G)`	retourne le *nombre* de sommets de `G`, c'est-à-dire la *taille* de la liste précédente
`sommetNom(G,etiquette)`	retourne le *sommet* de `G` désigné par son *nom* (`etiquette`), par exemple : `sommetNom (tgv2005, 'Bordeaux')`

2 Coloration

`colorierSommet(s,c)`	colorie le *sommet* `s` avec la *couleur* `c` (par exemple `'red'`)
`couleurSommet(s)`	retourne la *couleur* du *sommet* `s`.
`dessiner(G)`	dessine le *graphe* `G`

On peut modifier la couleur d'un sommet avec un appel à la fonction `colorierSommet(s,c)`, où c est une simple chaîne de caractères. La fonction `dessiner(G)` tient compte des couleurs des sommets si celles-ci font partie d'une liste prédéfinie ; par exemple `'red', 'green', 'blue'` sont des couleurs reconnues par le programme de dessin, et la liste complète se trouve à l'adressei `http://www.graphviz.org/doc/info/colors.html` .

Question 2.1 *Écrire une fonction* ***`toutColorier(G,c)`*** *qui colorie tous les sommets du graphe G avec la couleur c. Vous pouvez vérifier le résultat en affichant la liste des sommets du graphe, ou en le dessinant.*

Question 2.2 *Écrire une fonction* ***`existeCouleur(G,c)`*** *qui renvoie* ***`True`*** *s'il existe au moins un sommet de couleur c dans le graphe G et* ***`False`*** *sinon.*

3 Voisins

Deux sommets s et t sont dits *voisins* s'il existe une arête e ayant s et t comme extrémités ; on dit que l'arête e est *incidente* à chacun des sommets s et t.

La fonction `listeVoisins(s)` retourne la liste des voisins du sommet s, obtenue en suivant chacune des arêtes incidentes. Un même sommet peut se retrouver plusieurs fois dans cette liste. Une boucle autour du sommet s est par convention deux fois incidente à s : la liste des voisins d'un sommet portant une boucle contient deux fois le sommet lui-même, à cause de la boucle que l'on peut considérer dans un sens ou dans l'autre.

Le *degré* d'un sommet s est le nombre d'incidences d'arêtes, et la fonction `degre(s)` calcule sa valeur.

`listeVoisins(s)`	retourne la *liste* des *voisins* du sommet `s`
`degre(s)`	retourne le *degré* du sommet `s`, qui est aussi la *taille* de la liste précédente

Question 3.1 *Écrire une fonction* `sontVoisins(s1,s2)` *qui teste si les sommets* $s1$ *et* $s2$ *sont voisins, c'est-à-dire qui renvoie* `True` *si c'est le cas et* `False` *sinon.*

Question 3.2 *Écrire une fonction* `listeVoisinsCommuns (s1,s2)` *qui calcule la liste des voisins communs à deux sommets* $s1$ *et* $s2$.