

Initiation à l'informatique - 1er semestre

Les fonctions de manipulation de graphes

Le tableau ci-dessous décrit les fonctions déjà écrites dont vous vous servirez pour programmer. Pour les utiliser, il faut écrire en début de fichier la ligne :

```
from bib import *
```

La première colonne du tableau donne le nom de la fonction et des paramètres. Il faut respecter strictement l'orthographe de ce nom de fonction. Les noms des paramètres peuvent par contre changer. Par exemple, pour colorier en rouge un sommet mémorisé dans une variable qui s'appelle `sommet1`, on utilisera l'instruction `colorier(sommet1,"red")`. Notez qu'il n'y a pas de guillemets autour de `sommet1` (parce que c'est une variable), mais qu'il y en a autour de `"red"` (parce que c'est une constante chaîne de caractères). La deuxième colonne du tableau décrit ce que fait la fonction. Les fonctions des deux dernières lignes (en grisé) ne sont pas indispensables. Par contre, il est nécessaire de savoir manipuler toutes les autres fonctions.

Fonction et arguments	Fonctionnalité
<code>nom_graphe(G)</code>	Retourne le nom du graphe G.
<code>nb_sommets(G)</code>	Retourne le nombre de sommets du graphe G.
<code>liste_sommets(G)</code>	Retourne la liste des sommets du graphe G.
<code>chercher_sommet(G,nom)</code>	Retourne le sommet de G qui s'appelle <code>nom</code> , ou <code>None</code> (qui signifie « aucun ») s'il n'y a aucun sommet avec ce nom.
<code>nom_sommet(s)</code>	Retourne le nom du sommet <code>s</code> .
<code>degre(s)</code>	Retourne le degré du sommet <code>s</code> .
<code>liste_voisins(s)</code>	Retourne la liste des voisins du sommet <code>s</code> .
<code>marquer_sommet(s)</code>	Marque le sommet <code>s</code> .
<code>demarquer_sommet(s)</code>	Démarque le sommet <code>s</code> .
<code>est_marque(s)</code>	Retourne <code>True</code> si le sommet <code>s</code> est marqué, <code>False</code> sinon.
<code>colorier(s,c)</code>	Colorie le sommet <code>s</code> avec la couleur <code>c</code> . Les noms de couleurs sont des chaînes de caractères, en anglais. Les couleurs disponibles sont les suivantes : <code>"red"</code> , <code>"green"</code> , <code>"blue"</code> , <code>"orange"</code> , <code>"cyan"</code> , <code>"magenta"</code> , <code>"yellow"</code> , <code>"white"</code> .
<code>couleur(s)</code>	Retourne la couleur du sommet <code>s</code> .
<code>afficher(G)</code>	Affiche le graphe sous forme d'un texte.
<code>dessiner(G)</code>	Dessine le graphe dans une fenêtre séparée. Suppose que le logiciel <code>dot</code> est installé, http://www.graphviz.org (sous OS X, avec l'interface graphique, sous Linux, avec le logiciel <code>gv</code>).
<code>sommet(G,i)</code>	Retourne le sommet numéro <code>i</code> du graphe G, où <code>i</code> est un entier compris entre 0 et <code>nb_sommets(G)-1</code> .
<code>voisin(s,i)</code>	Autour d'un sommet <code>s</code> , les brins d'arêtes sont numérotés de 0 à <code>degre(s)-1</code> . La fonction retourne le sommet obtenu à partir de <code>s</code> en suivant l'arête numéro <code>i</code> , avec cette numérotation.

Tester la bibliothèque

Plusieurs graphes sont prédéfinis. Chaque graphe prédéfini est mémorisé dans une variable. Par exemple, le graphe de la figure 1 est accessible sous python via la variable `Konigsberg`. Testez les fonctions précédentes sur le graphe de la figure 1. Par exemple, pour afficher la liste de ses sommets, on tape sous l'interpréteur python la ligne suivante :

```
print liste_sommets(Konigsberg)
```

(après avoir tapé `from bib import *`) et l'interpréteur python répond [`<A,-,->`, `<B,-,->`, `<C,-,->`, `<D,-,->`]. Ceci indique qu'il y a 4 sommets appelés A, B, C et D. Le - qui suit le nom de chaque sommet indique que le sommet n'est pas colorié. Le - suivant indique que le sommet n'est pas marqué. De même, pour colorier en vert le sommet A de ce graphe :

```
s = chercher_sommet(Konigsberg,"A")
colorier(s,"green")
```

La première instruction mémorise dans la variable `s` le sommet A de `Konigsberg`. La seconde colorie ce sommet en vert, ce qui peut être visualisé en tapant à nouveau `print liste_sommets(Konigsberg)` qui affiche maintenant [`<A,green,->`, `<B,-,->`, `<C,-,->`, `<D,-,->`]

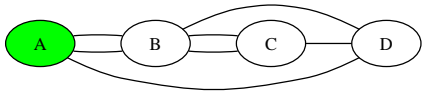


FIG. 2: Konigsberg vu de l'interpréteur

Pour dessiner le graphe, il suffit d'entrer l'instruction python `dessiner(Konigsberg)`. On visualise le dessin de la figure 2 ci-contre, différent de celui de la Figure 1, mais qui représente bien le même graphe.

Voisins

Chaque sommet est « touché » par un certain nombre d'extrémités d'arêtes (aussi appelées *brins d'arêtes*). Le nombre des extrémités d'arêtes touchant un sommet est le *degré* du sommet. Par exemple, sur le graphe de la figure 3, le sommet A a comme degré 2, car il est touché par 2 brins d'arêtes. Le sommet B a degré 4, car il est touché par 4 brins d'arêtes : deux qui le relient au sommet A, et une boucle qui le touche deux fois. Enfin, le sommet C a pour degré 0 et le sommet D a pour degré 4.

La liste retournée par la fonction `liste_voisins(s)` est celle des sommets atteints en suivant chacun des brins d'arêtes depuis `s`. Un même sommet `t` peut donc apparaître plusieurs fois comme voisin de `s`. Par exemple,

```
s = chercher_sommet(gr4_2,"B")
print liste_voisins(s)

affiche
[<B,-,->, <B,-,->, <A,-,->, <A,-,->].
```

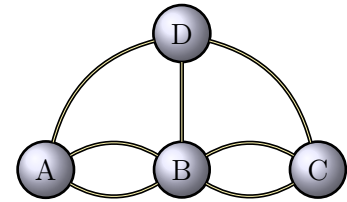


FIG. 1: Graphe Konigsberg

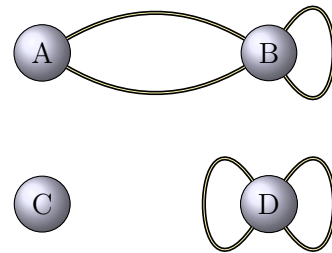


FIG. 3: Graphe gr4_2