

TP 2 - Parcours en largeur et en profondeur

Théorie et Algorithmique des Graphes
L3 INFO - Semestre 6

Le but de ce TP est de mettre en œuvre en Python les méthodes de parcours en largeur et en profondeur d'un graphe. Ce TP sera complété par le TP3, portant sur les applications de ces parcours. Rappelons que ces deux algorithmes s'appliquent indifféremment à un graphe orienté ou non orienté et devront donc être testés dans les deux cas. Dans la feuille de tests unitaires fournie avec le TP, les fonctions seront testées avec les graphes suivants :

- 3 graphes orientés :
 - $G_1 = \{1 : [5], 2 : [1,4], 3 : [2], 4 : [3], 5 : [2,4]\}$
 - $G_2 = \{1 : [5], 2 : [1,4,5], 3 : [2,4], 4 : [], 5 : [4]\}$
 - $G_3 = \{1 : [3,5,6], 2 : [1], 3 : [2,4], 4 : [], 5 : [], 6 : [4]\}$
- et le graphe de Petersen :
 - Petersen = {
 - 1 : [2,5,6], 2 : [1,3,7], 3 : [2,4,8], 4 : [3,5,9], 5 : [1,4,10],
 - 6 : [1,8,9], 7 : [2,9,10], 8 : [3,6,10], 9 : [4,6,7], 10 : [5,7,8]

1. Parcours en largeur

- On utilise un ensemble `Visite` dans lequel on ajoute les sommets visités pour ne pas les traiter 2 fois.
- On utilise une file `File` pour gérer la file d'attente des sommets à visiter prochainement.
- Le résultat de l'appel de la fonction sera la liste `ordreVisite` des sommets visités dans l'ordre de ce parcours.

Écrire les fonctions suivantes :

1. `largeur(G,i)` : effectue le parcours en largeur du graphe G à partir du sommet i
2. `largeurG(G)` : effectue le parcours en largeur généralisé à tout le graphe G

2. Parcours en profondeur

- On utilise toujours l'ensemble `Visite` défini dans le parcours en largeur.
- Le résultat de l'appel de la fonction sera la liste `ordreVisite` des sommets visités dans l'ordre de ce parcours. En fonction de l'endroit où on positionne la mise à jour de la liste `ordreVisite` la fonction donnera l'ordre de première visite ou l'ordre de dernière visite. On pourra passer d'une version à l'autre en commentant une des lignes de mise à jour.

Comme souvent lorsqu'on écrit un algorithme récursif, on distinguera une fonction auxiliaire (récursive) et une fonction d'appel dont le rôle est d'initialiser les variables et d'enclencher le premier appel récursif.

Écrire les fonctions suivantes :

1. `profRec(G,i,Visite,ordreVisite)` : fonction auxiliaire récursive qui provoque un parcours en profondeur du graphe à partir du sommet i . Cette fonction ne retourne aucun résultat et se contente de mettre à jour les paramètres `Visite` et `ordreVisite`.
2. `profond(G,i)` : effectue le parcours en profondeur du graphe G à partir du sommet i
3. `profondG(G)` : effectue le parcours en profondeur généralisé à tout le graphe G .