

Numérique et Sciences Informatiques
Chapitre V - Structures de données relationnelles
Travaux Dirigés 11

Dans ce TD, on utilisera les deux graphes suivants :

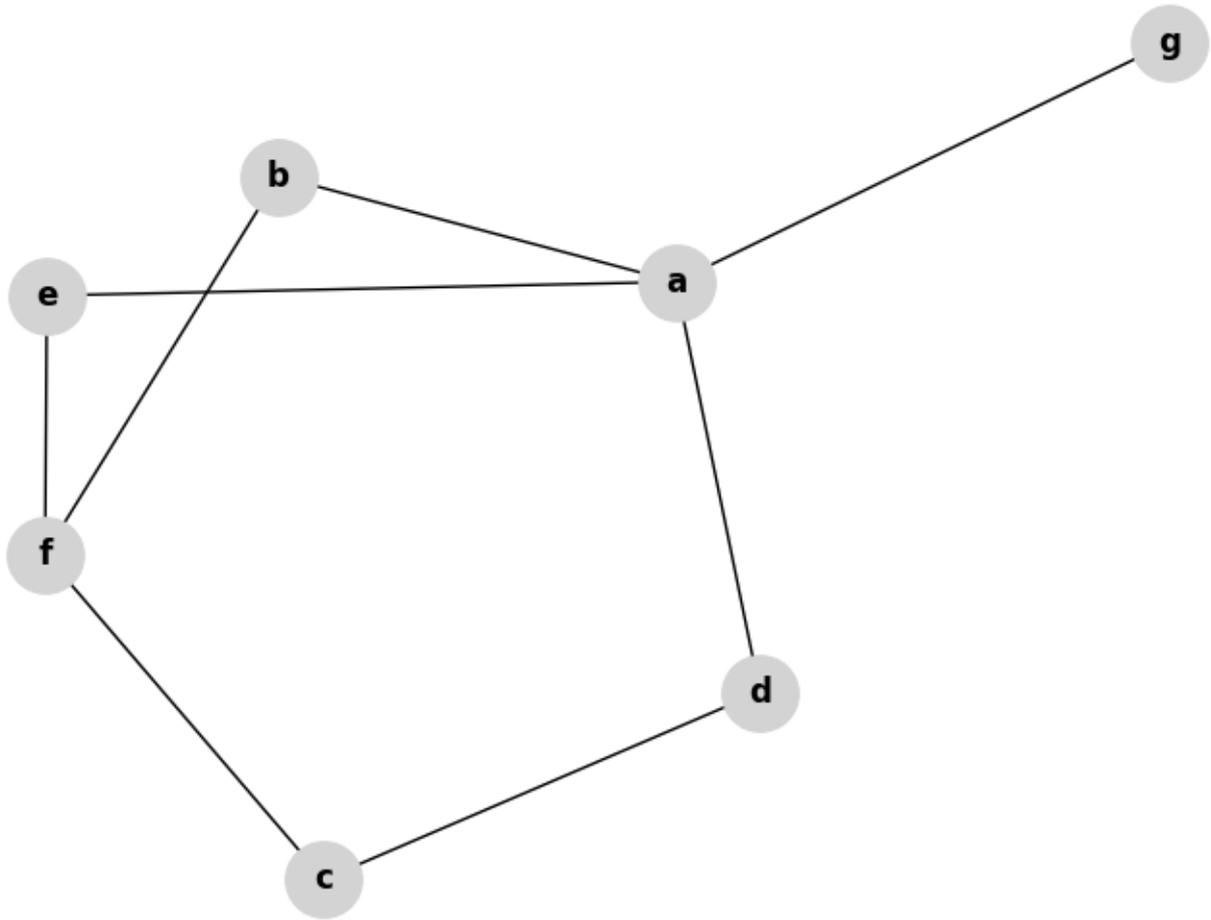


FIGURE 1 – graphe 1

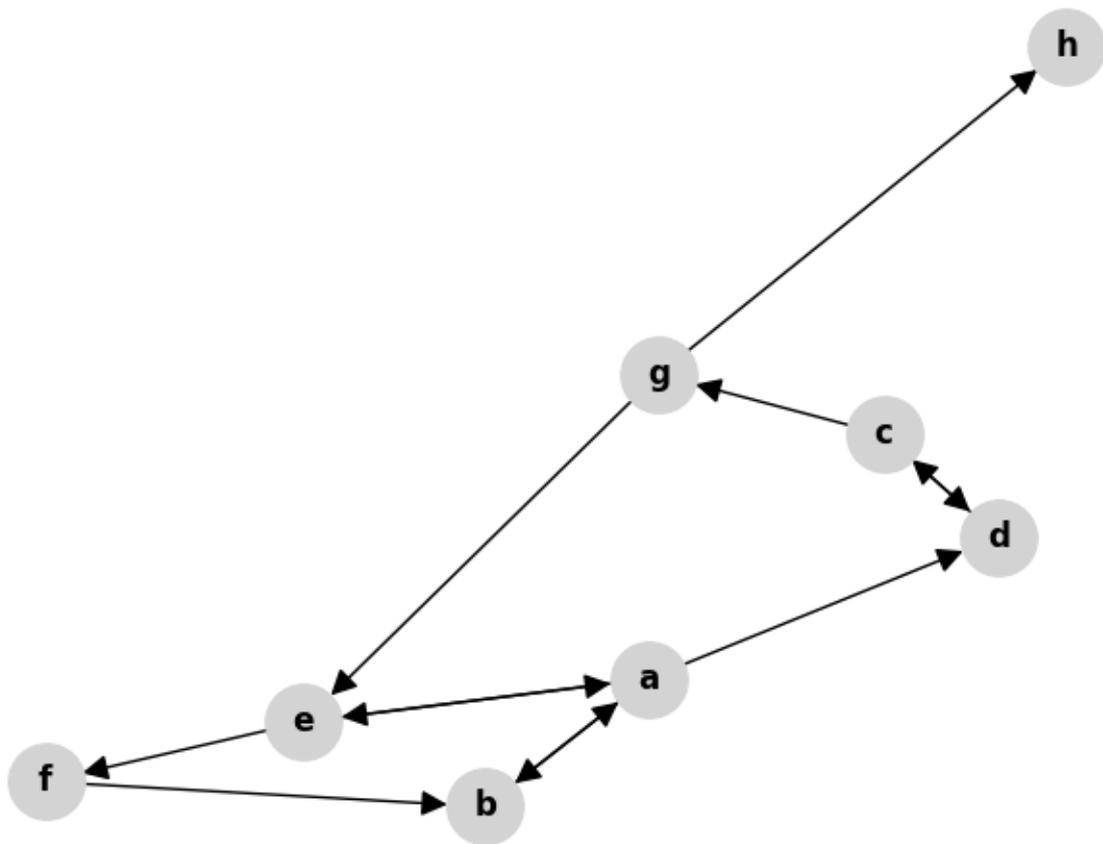


FIGURE 2 – graphe 2

I. Parcours en profondeur d'abord

Pour parcourir un graphe en profondeur d'abord :

- On explore les sommets du graphe en passant de sommet en sommet en suivant l'un des voisins(ou successeurs) et en marquant les sommets visités.
 - Lorsqu'il n'y a plus de sommets accessibles non encore visités, on revient au sommet précédent.
1. Tester cet algorithme sur les deux graphes en commençant par le sommet a .
 2. Quelle structure de données permet d'implémenter la liste des voisins des sommets précédents qui n'ont pas été visités lors du parcours en profondeur d'abord ?
 3. Implémenter une fonction `parcours_profondeur(graphe, sommet, sommets_visites = None, sommets_non_visites = None)` qui prend en paramètre un graphe, un sommet du graphe, la liste des sommets visités (qui prendra comme valeur par défaut `None`) et la "liste" des sommets non visités ((qui prendra comme valeur par défaut `None`)) et qui renvoie un tableau des sommets visités dans l'ordre d'un parcours en profondeur d'abord.

II. Parcours en largeur d'abord

Pour parcourir un graphe en largeur d'abord :

- On explore les sommets du graphe en explorant d'abord tous les voisins (ou successeurs) d'un sommet.
 - Ensuite on explore les voisins des voisins(ou successeurs des successeurs).
1. Quelle structure de données permet d'implémenter la liste des voisins des sommets précédents qui n'ont pas été visités ?
 2. Implémenter une fonction `parcours_largeur(graphe, sommet, sommets_visites = None, sommets_non_visites = None)` qui prend en paramètre un graphe, un sommet du graphe, la liste des sommets visités (qui prendra comme valeur par défaut `None`) et la "liste" des sommets non visités ((qui prendra comme valeur par défaut `None`)) et qui renvoie un tableau des sommets visités dans l'ordre d'un parcours en largeur d'abord.

III. Recherche dans un graphe

III.1. Recherche d'une chaîne (ou d'un chemin)

Implémenter une fonction `recherche_chemin(graphe, sommet1, sommet2)` qui prend en paramètre un graphe et deux sommets différents et qui renvoie une chaîne (un chemin) sous forme d'un tableau de sommets, s'il existe, et `None` sinon.

III.2. Présence d'un cycle (ou d'un circuit)

Implémenter une fonction `presence_cycle(graphe, sommet)` qui prend en paramètre un graphe et un sommet et qui renvoie `True` si un cycle (ou circuit) existe dans le graphe, `False` sinon.