

Parcours de graphes

Après avoir introduit la notion de graphes, nous allons voir maintenant comment on peut utiliser ces structures .

Nous allons étudier le parcours en largeur, en profondeur d'un graphe, rechercher un cycle ou un certain chemin.

Quelques définitions :

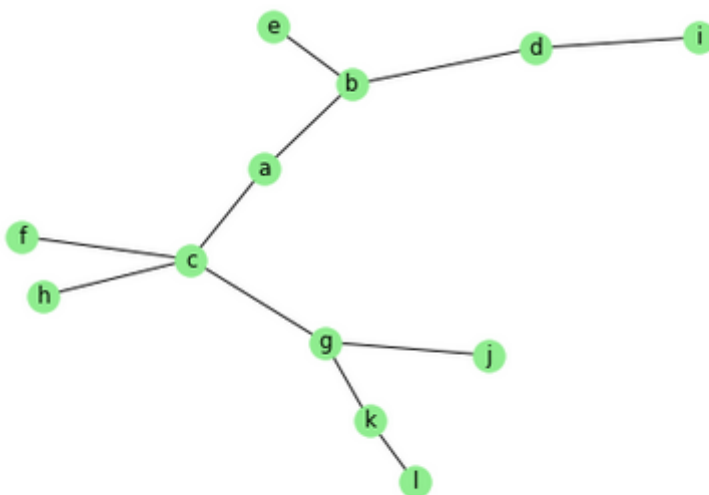
- On appelle distance entre deux sommets le nombre d'arêtes minimum pour aller d'un sommet à l'autre
- On appelle écartement d'un sommet la distance maximale entre ce sommet et les autres sommets du graphe.
- Le centre d'un graphe est le sommet d'écartement minimal (non nécessairement unique)
- Le rayon d'un graphe est l'écartement du centre du graphe
- Le diamètre d'un graphe est la distance maximale entre deux sommets

1. Parcours en largeur d'abord (Breadth first search)

Parcourir un graphe veut dire visiter tous ses sommets et toutes ses arêtes.

Il faut bien sur un peu de méthode pour être exhaustif.

Graphe 1



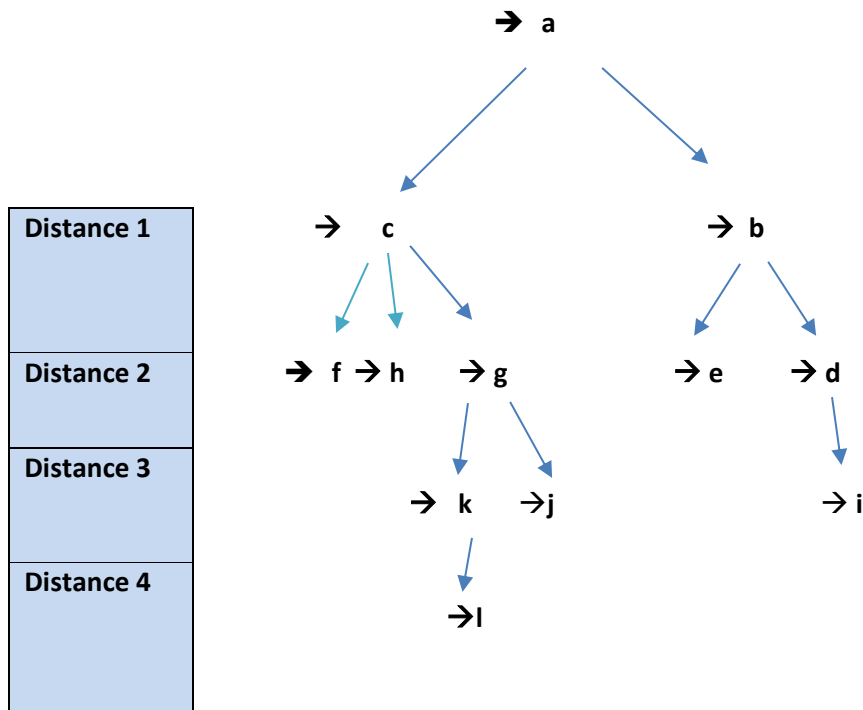
Parcourir en largeur un graphe à partir d'un sommet revient à visiter les voisins de ce sommet (i.e. les sommets à distance 1) puis les enfants des voisins (les sommets à distance 2 du point de départ) etc....

Q1. Donner l'écartement du sommet a et la distance entre les sommets a et j.

Q2. Donner le centre, le rayon et le diamètre de ce graphe.

Parcours en largeur depuis le sommet a.

Graphe 2



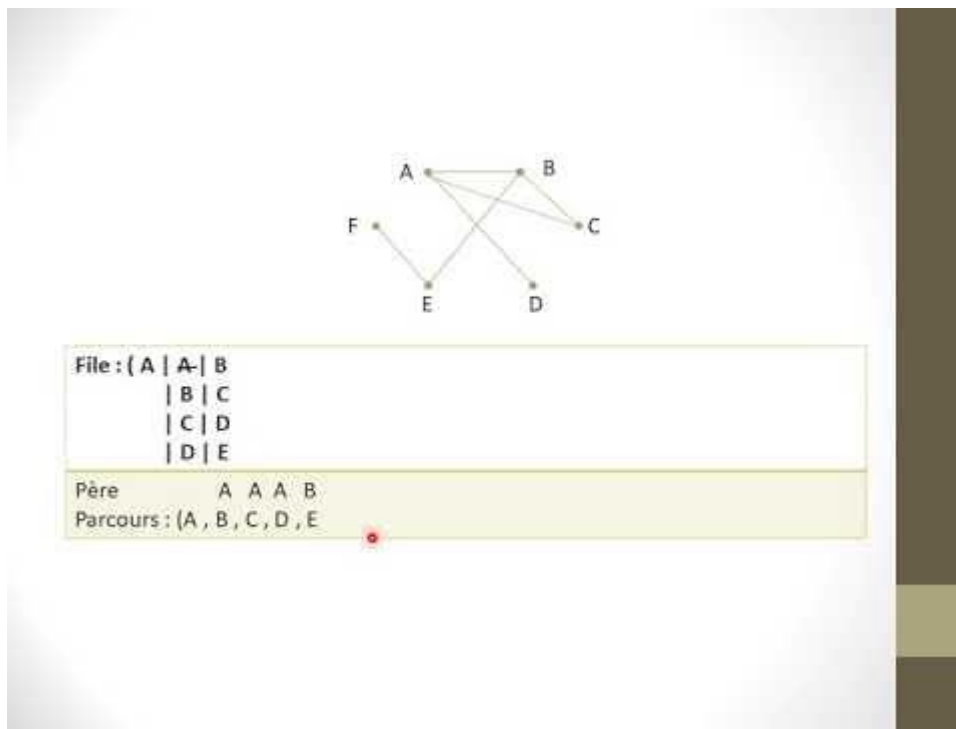
Q3. Réaliser le même parcours en largeur en partant d'un autre sommet.

BFS et algorithme :

Nous allons écrire un algorithme qui parcourt en largeur un graphe à partir d'un sommet choisi.

Principe de l'algorithme. On va utiliser une file pour gérer les sommets (nœud). Les files sont du type « FIFO ».

1. On enfile le sommet de départ
2. On enfile les voisins (à distance 1 ou sommets adjacents au nœud initial) s'ils n'ont pas été visités.
3. On défile le sommet de départ
4. On réitère les points 2 et 3 tant que la file n'est pas vide.



2. Parcours en profondeur d'abord (Depth first search)

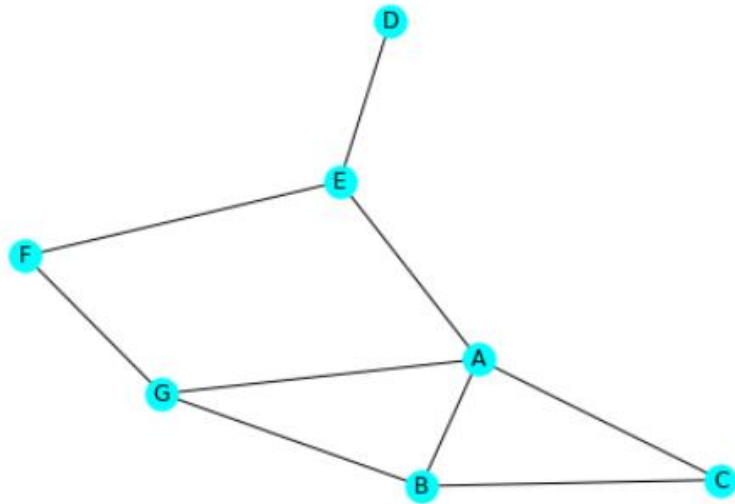
Contrairement au parcours en largeur, on va s'intéresser ici au parcours du graphe branche par branche. Dès que l'on s'engage sur une branche, on va le plus loin possible. Une fois fini, on passe à la branche suivante en « remontant ».

DFS et algorithme :

1. On place le sommet de départ dans la pile.
2. On met les voisins du sommet de départ dans la pile s'ils n'ont pas été visités.
3. On dépile le dernier sommet entré.
4. On recommence tant que la pile n'est pas vide.

Avec cette méthode, voici un résultat possible du parcours en profondeur du graphe suivant :

Graphe 3



Etat de la pile	Etat des nœuds visités
A	A
B C G E	A E
B C G F D	A E D
B C G F	A E D F
B C G G	A E D F G
B C B	A E D F G B
B C	A E D F G B C
Vide	L'algo s'arrête.

Etablir un parcours en profondeur de ce graphe en partant d'un autre sommet.
Ecrire un algorithme de parcours en profondeur.

| F
 | C | E
 | B | B | B
 Pile : (A | A | A | A
 Père A B B E
 Parcours : (A , B , C , E , F

3. Recherche d'un cycle dans un graphe

Définitions :

- Une chaîne est une suite d'arêtes consécutives dans un graphe. Elle est désignée par la suite des sommets par lesquels elle passe.
Dans le graphe précédent, A-E-D est une chaîne.
- Un cycle est une chaîne qui commence et finit par le même sommet et qui ne passe pas plusieurs fois par la même arête. On parlera de circuit pour un graphe orienté.

Parmi les trois graphes du cours, citez ceux qui possèdent un cycle. Citez alors un cycle de longueur 3, 4, 5.

Quel est le cycle de longueur minimale, maximale ?

Cycle et algorithme

Pour déterminer si un graphe possède un cycle, il nous faut « plus » que le parcourir. Ne devant pas repasser par une arête déjà visitée, il nous faut répertorier les parents de chaque sommet visité.

On va utiliser un parcours en largeur d'abord que l'on va « aménager » pour répondre à notre problématique.

1. On place le sommet de départ dans la file
2. On met les voisins du sommet de départ dans la file s'ils n'ont pas été visités.
3. On défile le premier sommet entré : S'il a été déjà visité, on renvoie vrai, sinon on réitère l'opération jusqu'à ce que la file soit vide (on renvoie faux) ou que l'on renvoie vrai.

Exercice 1

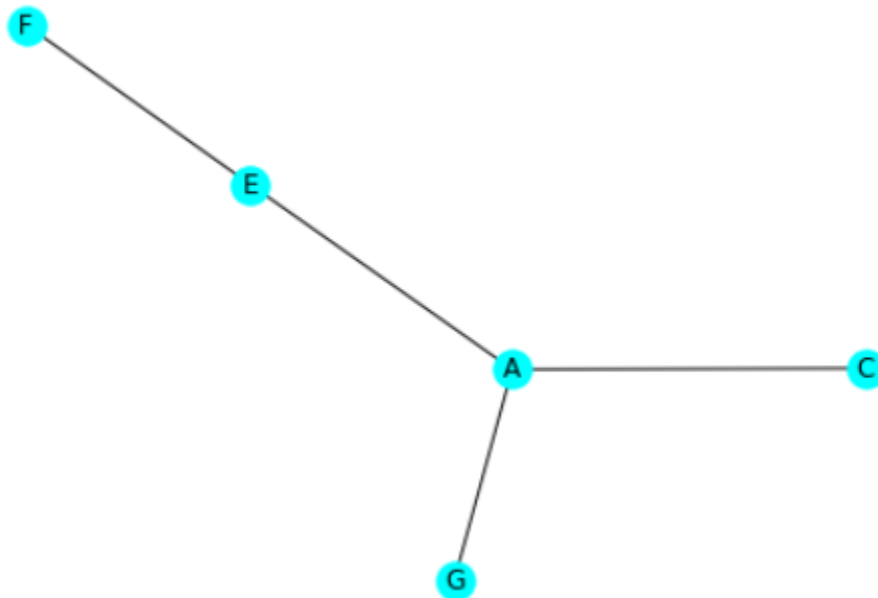
Appliquer cet algorithme avec le graphe 3 en partant de E

Etat de la file avant analyse	Nœuds visités
E	E
A D F	E A
D F C B G	E A D
F C G B G	E A D F
C G B G B	E A D F C
G B G B B	E A D F C G
B G B B: Algorithme stoppé : Nœud déjà dans visite	E A D F C G B True

Exercice 2

Appliquer cet algorithme avec le graphe 4 ci-dessous, en partant du sommet de votre choix.

Graphe 4



Etat de la file avant analyse	Nœuds visités
E	E
F A	E F
A	E F A
A G C	E F A G
G C	E F A G C
C	False

Exercice 3.

Réécrire l'algorithme en utilisant un parcours en profondeur pour chercher la présence d'un cycle. L'appliquer sur l'exercice 1.

4. Recherche d'un chemin dans un graphe

La recherche d'un chemin est un cas général de la recherche d'un cycle. Connaissant un nœud de départ et un nœud d'arrivée, peut-on trouver un chemin qui relie ces deux nœuds ?

Exercice :

Adaptez l'algorithme de recherche d'un cycle pour trouver un chemin entre deux nœuds donnés.