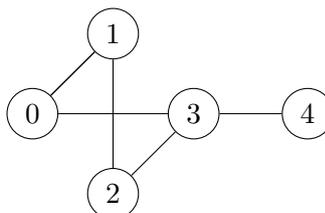


Informatique - DS 5

Proposition de corrigé

Exercice 1 : Liste et matrice d'adjacence

On considère le graphe G suivant :



1. Ce graphe est-il connexe ? Justifier.

Le graphe G est connexe car pour toute paire de sommets, il existe un chemin le reliant (le graphe est en « un seul morceau »).

2. Donner, en justifiant, l'ordre et la taille du graphe G , ainsi que le degré du sommet 3.

Le graphe G est d'ordre 5 car il comporte 5 sommets, et de taille 5 aussi car il comporte 5 arêtes.

Le sommet 3 est de degré 3 car il a 3 arêtes incidentes.

3. Donner, sans justification, la liste d'adjacence L du graphe G .

(on la donnera sous la forme d'une liste de listes Python : `[[...], [...], ...]`)

$L = [[1, 3], [0, 2], [1, 3], [0, 2, 4], [3]]$

4. Donner, sans justification, la matrice d'adjacence A du graphe G .

(on la donnera sous la forme d'une liste de listes Python : `[[...], [...], [...]]`)

$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

5. Par lecture du graphe, donner (en listant leurs sommets) les chaînes de longueur 3 reliant les sommets 2 et 3. Combien y en a-t-il ?

2-1-0-3 2-1-2-3 2-3-2-3 2-3-0-3 2-3-4-3. Il y en a 5.

6. On considère le script Python ci-contre.
Que font les première et troisième lignes ?

```
import numpy.linalg as al
def f(M,k):
    N=al.matrix_power(M, k)
    return N
```

La première ligne importe la sous-bibliothèque `linalg` de la bibliothèque `numpy` pour accéder aux instructions d'algèbre linéaire.

Le préfixe `numpy.linalg` est remplacé par `al`.

La troisième ligne appelle l'instruction `matrix_power` de la sous-bibliothèque `linalg` qui prend en entrée une matrice carrée M et l'entier k , élève la matrice à la puissance k , et affecte à la variable N la matrice M^k .

7. On suppose que l'on a saisi la matrice A et on considère les instructions :

```
B=f(A, 2)
n=B[2][3]
print(n)
```

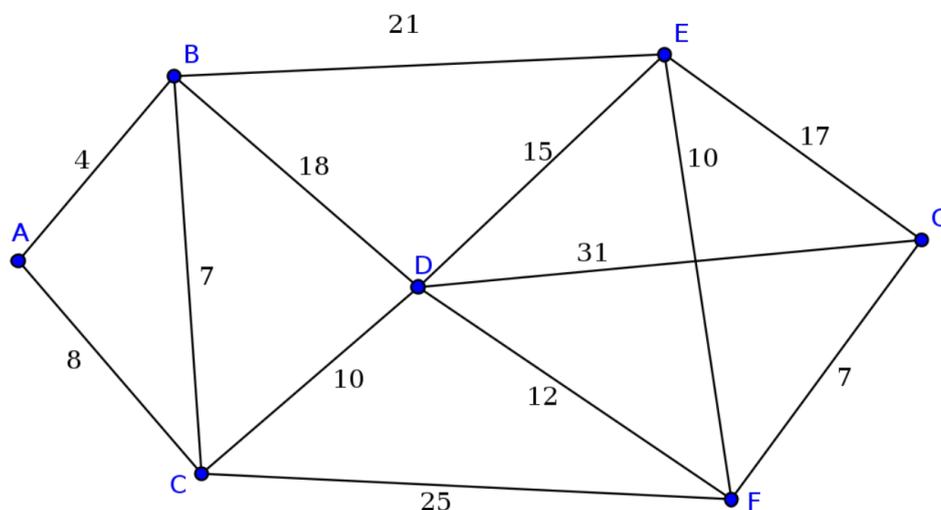
Compléter ces instructions pour qu'elles permettent l'affichage du nombre trouvé à la question 5.

8. Écrire la fonction `Trois` qui prend en entrée la matrice A et renvoie le nombre total de chemins de longueur 3 dans le graphe G .

```
1 def Trois(M):
    P=f(M,3)
3     s=0
    for i in range(5):
5         for j in range(5):
            s+=P[i][j]
7             print(s)
    return s
```

Exercice 2 : Algorithme de Dijkstra

Une région est munie d'un réseau de trains, représenté par le graphe ci-dessous.



Les stations sont symbolisées par les sommets A, B, C, D, E, F et G.

Chaque arête représente une ligne reliant deux gares. Les temps de parcours en minutes entre chaque sommet sont indiquées sur le graphe.

1. Dans le tableau ci-dessous, mettre en oeuvre l'algorithme de Dijkstra prenant comme point de départ le sommet B et point d'arrivée le sommet G.

| A | B | C | D | E | F | G |
|----------|---|----------|----------|----------|----------|----------|
| ∞ | 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 4,B | x | 7,B | 18,B | 21,B | ∞ | ∞ |
| x | x | 7,B | 18,B | 21,B | ∞ | ∞ |
| x | x | x | 17,C | 21,B | 32,F | ∞ |
| x | x | x | x | 21,B | 29,D | 48,D |
| x | x | x | x | x | 29,D | 38,E |
| x | x | x | x | x | x | 36,F |

2. En déduire le plus court chemin en minutes reliant la gare B à la gare G.
(liste des gares et durée totale).

Le plus court chemin est B-C-D-F-G et dure 36 minutes.

~