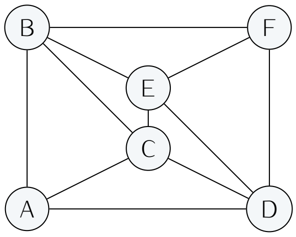
Exercices du chapitre 24
Graphes

Exercice 1 - ●○○

On considère le graphe $\mathscr{G}$ suivant :



1. Donner la matrice d'adjacence M de ce graphe.
2. L'exécution du programme ci-dessous affiche le résultat suivant.

```python
import numpy as np

M=np.array([[0,1,1,1,0,0],[1,0,1,0,1,1],
            [1,1,0,1,1,0],[1,0,1,0,1,1],
            [0,1,1,1,0,1],[0,1,0,1,1,0]])
print(np.dot(M,M))
```

```
[[3 1 2 1 3 2]
 [1 4 2 4 2 1]
 [2 2 4 2 2 3]
 [1 4 2 4 2 1]
 [3 2 2 2 4 2]
 [2 1 3 1 2 3]]
```

2.a. Combien existe-t-il de chaînes de longueur 2 dont le départ et l'extrémité sont respectivement les sommets B et D ? Donner ces chaînes.
2.b. Justifier que le graphe $\mathscr{G}$ est connexe.

3. Ce graphe possède-t-il une chaîne eulérienne ? Si oui, en donner une.
4. Justifier que ce graphe ne possède pas de cycle eulérien. Quelle arête ajouter pour qu'il en possède un ?
5. Expliquer ce que permet de renvoyer l'exécution de **mystere(M)** dans le cas où **M** est le tableau **numpy** représentant la matrice d'adjacence du graphe $\mathscr{G}$.

```python
def mystere(M):
    S=['A','B','C','D','E','F']
    n=len(S)
    L=[[] for k in range(n)]
    for i in range(0,n):
        for j in range(0,n):
            if M[i,j]!=0:
                L[i].append(S[j])
    return L
```

Exercice 2 - ●●○ - Vrai / faux

1. Un graphe est connexe si, et seulement si, il n'a aucun sommet isolé.
2. Dans un graphe simple et connexe, il existe toujours au moins deux sommets de même degré.
3. Soit $\mathscr{G}$ un graphe simple, non orienté et connexe. Démontrer que si $\mathscr{G}$ ne possède pas de cycle eulérien, alors il est possible d'ajouter un unique sommet à $\mathscr{G}$ de sorte que le nouveau graphe obtenu possède un cycle eulérien.

Exercice 3 - ●●○ - Connexité

Soit $\mathscr{G}$ un graphe non orienté, simple, d'ordre $2p$ tel que le degré de chaque sommet est supérieur ou égal à p.
Démontrer que $\mathscr{G}$ est connexe.

Exercice 4 - ●●○ - Connexité (bis)

On considère un graphe $\mathscr{G}$ simple et non orienté d'ordre $n \geqslant 2$.

1. Écrire une fonction **Python** prenant en argument d'entrée la matrice d'adjacence d'un tel graphe $\mathscr{G}$ et renvoyant un booléen précisant la connexité de $\mathscr{G}$.
2. On suppose que $\mathscr{G}$ est connexe et possède un sommet x de degré 1. Montrer que $\mathscr{G} \setminus \{x\}$ est connexe.
3. En déduire que si $\mathscr{G}$ est connexe d'ordre n, alors $\mathscr{G}$ possède au moins $n-1$ arêtes.
4. Comment utiliser le résultat précédent pour effectuer un test de non connexité d'un graphe simple et non orienté qui soit peu coûteux en calculs ?

Exercice 5 - ●●○ - Tournoi

On appelle **tournoi** un graphe orienté et simple tel qu'entre deux sommets distincts, il existe un et un seul arc.
On dit qu'un sommet est un roi lorsqu'il a un chemin orienté de longueur au plus 2 vers tous les autres sommets.

1. Combien un tournoi contient-il d'arcs ?
2. Donner un tournoi d'ordre 4 possédant un unique roi.
3. Montrer que, dans un tournoi, il existe toujours un roi.
4. 4.a. Que permet de faire la fonction **mystere** suivante, où **L** désigne une liste de réels ?

```python
def mystere(L):
    m=L[0]
    k=0
    for i in range(1,len(L)):
        if L[i]>m:
            m=L[i]
            k=i
    return k
```

   4.b. Proposer une fonction **Python** prenant en argument d'entrée la matrice d'adjacence d'un tournoi dont les sommets sont numérotés à partir de 0 et renvoyant en sortie un roi.

Exercice 6 - ●●● - Graphe régulier

On dit qu'un graphe est **régulier** lorsqu'il est simple, non orienté et que tous ses sommets ont même degré.

1. Écrire une fonction **Python** prenant en argument d'entrée la matrice d'adjacence d'un graphe simple non orienté et renvoyant **True** s'il est régulier, **False** sinon.
2. Que dire de l'ordre d'un graphe régulier dont les sommets sont tous de degré 3 ?
3. Démontrer que pour tout $p \in [\![2;+\infty[\![$, il existe un graphe régulier d'ordre $2p$ dont les sommets sont de degré 3.

Exercice 7 - ●●● - Graphes aléatoires

Toutes les variables aléatoires de cet exercice sont supposées définies sur le même espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$. Soit n un entier naturel supérieur ou égal à 2. On considère un graphe simple non orienté d'ordre n dont les sommets sont numérotés de 1 à n. Les arêtes de ce graphe sont définies de façon aléatoire et indépendantes entre elles de la sorte : pour tous i, j de $[\![1;n]\!]$ différents, on construit une arête entre les sommets i et j avec une probabilité $p \in]0;1[$.
Pour tout $i \in [\![1;n]\!]$, on définit une variable aléatoire X_i telle que :

$$X_i = \begin{cases} 1 & \text{si le sommet } i \text{ est isolé} \\ 0 & \text{sinon} \end{cases}$$

Enfin, on pose $S_n = \displaystyle\sum_{i=1}^{n} X_i$.

1. Écrire une fonction **Python** prenant en arguments un entier $n \geqslant 2$ et un réel $p_n \in]0;1[$ puis qui génère et renvoie la matrice d'adjacence d'un tel graphe aléatoire.
2. Que représente la variable aléatoire S_n ?
3. Interpréter le résultat renvoyé par l'exécution de **mystere(M)** où M désigne une matrice générée par le programme de la question 1..

```python
import numpy as np

def mystere(M):
    L=[]
    (n,n)=np.shape(M)
    for i in range(0,n):
        if sum(M[i,:])==0:
            L.append(i+1)
    return L,len(L)
```

4. Donner la loi de X_1. En déduire l'espérance de S_n.
5. Les variables aléatoires $X_1, \dots, X_n$ sont-elles mutuellement indépendantes ?