

≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈

Exercice : (4 points)

Somme et factoriel

1 pt **Q1-** Écrire la fonction **factoriel(k)** qui reçoit en paramètre un entier positif **k** et qui renvoie la valeur du factoriel de **k** : $k! = 1 * 2 * 3 * \dots * k$.

Exemples :

- La fonction **factoriel(5)** renvoie le nombre : $120 = 1 * 2 * 3 * 4 * 5$
- La fonction **factoriel(0)** renvoie le nombre : **1**

0.5 pt **Q2-** Déterminer la complexité de la fonction **factoriel(k)**, et justifier votre réponse.

1.5 pt **Q3-** Écrire la fonction **som_fact(L)** qui reçoit en paramètre une liste **L** de nombres entiers positifs. La fonction renvoie la somme des factoriels des éléments de **L**.

Exemple :

L = [5, 3, 0, 6, 1]

La fonction **som_fact(L)** renvoie la valeur de la somme : $5! + 3! + 0! + 6! + 1!$

1 pt **Q4-** Déterminer la complexité de la fonction **som_fact(L)**, et justifier votre réponse.

≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈ ≈

Problème :

Réseau routier

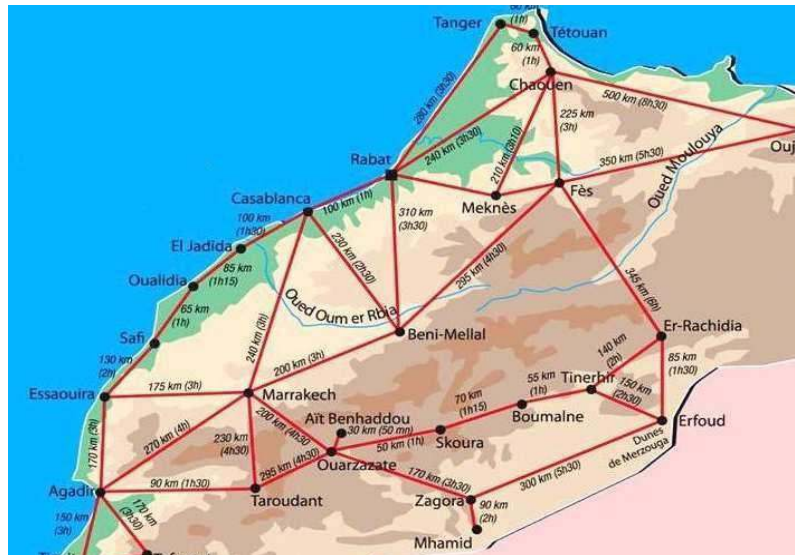


Figure 1 : Extrait du réseau routier du Maroc

Un réseau routier peut être représenté par un dessin qui se compose de points et de traits continus reliant deux à deux certains de ces points : les **points** sont les **villes**, et les **lignes** sont les **routes**. On considère que toutes les routes sont à double sens. Chaque route peut être affectée par une valeur qui peut représenter le temps ou la distance entre deux villes, ...

Étant donné un réseau routier, on pourra s'intéresser à la résolution des problèmes suivants :

- Quel est le plus court chemin entre deux villes ?
- Entre deux villes, quel est le nombre de chemins passant par un nombre de routes donné ?
- Est-il possible de passer par toutes les villes du réseau, sans passer deux fois par une même ville ?, si oui, quel est le plus court chemin ?
- Entre deux villes, quel est le chemin ayant le moindre coût ?
- ...

Partie I :

Modélisation d'un réseau routier

On considère un réseau routier composé de n villes (avec $n \geq 2$). Les villes du réseau routier sont numérotées par des entiers allant de 0 à $n-1$.

Exemple :

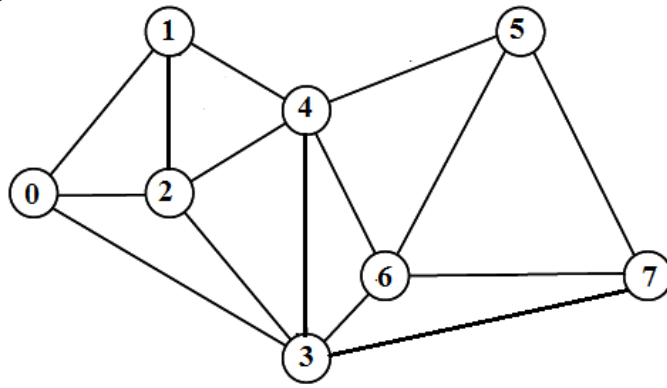


Figure 2 : Réseau routier composé de **15** routes, et de **8** villes numérotées de **0** à **7**.

Pour plus de clarté, tous les exemples de ce problème seront appliqués sur le réseau routier de la **figure 2**.

Représentation du réseau routier

Pour représenter un réseau routier de **n** villes, on utilise une matrice symétrique **R** d'ordre **n** (**n** lignes et **n** colonnes), telle que :

Pour toutes les villes **i** et **j**, telles que **0 ≤ i < n** et **0 ≤ j < n**, on a :

- $R[i, j] = R[j, i] = 1$, s'il existe une route qui relie entre la ville **i** et la ville **j** ;
- $R[i, j] = R[j, i] = 0$, sinon.

Exemple :

Le réseau routier de la **figure 2** est représenté par la matrice symétrique **R**, d'ordre **8**, suivante :

<i>i/j</i>	0	1	2	3	4	5	6	7
0	0	1	1	1	0	0	0	0
1	1	0	1	0	1	0	0	0
2	1	1	0	1	1	0	0	0
3	1	0	1	0	1	0	1	1
4	0	1	1	1	0	1	1	0
5	0	0	0	0	1	0	1	1
6	0	0	0	1	1	1	0	1
7	0	0	0	1	0	1	1	0

NB : Dans la matrice symétrique **R**, les lignes **i** et les colonnes **j** représentent les villes du réseau routier.

I. 1- Représentation de la matrice du réseau routier en Python

Pour représenter la matrice symétrique **R** d'ordre **n**, on utilise une liste composée de **n** listes qui sont toutes de même longueur **n**.

Exemple :

La matrice symétrique **R**, du réseau routier de la **figure 2**, est représentée par la liste **R**, composée de **8** listes, de taille **8** chacune :

```
R = [ [0,1,1,1,0,0,0,0], [1,0,1,0,1,0,0,0], [1,1,0,1,1,0,0,0],  
      [1,0,1,0,1,0,1,1], [0,1,1,1,0,1,1,0], [0,0,0,0,1,0,1,1],  
      [0,0,0,1,1,1,0,1], [0,0,0,1,0,1,1,0]  
      ]
```

R[i][j] est l'élément de **R**, à la **i^{ème}** ligne et la **j^{ème}** colonne.

Exemples : **R[0][0]** est l'élément : **0**, et **R[0][2]** est l'élément : **1**

R[i] est la ligne d'indice **i** dans **R**.

Exemple : **R[0]** est la liste **[0 , 1 , 1 , 1 , 0 , 0 , 0 , 0]**, qui représente la ligne d'indice **0** dans la matrice **R**.

Q 1- À partir de la matrice symétrique **R** de la **figure 2**, donner les résultats des expressions suivantes :

R[4][2] , **R[1]** , **len(R[2])** , **len(R)**

I. 2- Villes voisines

***i** et **j** sont deux villes dans un réseau routier représenté par une matrice symétrique **R**.*

*Les villes **i** et **j** sont **voisines**, s'il existe une route entre la ville **i** et la ville **j**.*

Q 2. a- Écrire la fonction **voisines (i, j, R)**, qui reçoit en paramètres deux villes **i** et **j** d'un réseau routier représenté par la matrice symétrique **R**. La fonction renvoie **True** si les villes **i** et **j** sont voisines, sinon, la fonction renvoie **False**.

Exemples :

- La fonction **voisines (3, 0, R)** renvoie **True** ; *(les villes 3 et 0 sont voisines)*
- La fonction **voisines (3, 5, R)** renvoie **False**. *(les villes 3 et 5 ne sont pas voisines)*

Q 2. b- Écrire la fonction **list_voisines (i, R)**, qui reçoit en paramètres une ville **i** d'un réseau routier représenté par la matrice symétrique **R**. La fonction renvoie la liste de toutes les villes voisines à la ville **i**.

Exemple :

La fonction **list_voisines (3, R)** renvoie la liste des villes voisines à la ville **3** : **[0 , 2 , 4 , 6 , 7]**

I. 3- Degré d'une ville

*Dans un réseau routier, le **degré** d'une ville **i** est le nombre de villes voisines à la ville **i**.*

Q 3- Écrire la fonction **degre (i, R)**, qui reçoit en paramètres une ville **i** d'un réseau routier représenté par la matrice symétrique **R**. La fonction renvoie le degré de la ville **i**.

Exemples :

- La fonction **degre (3, R)** renvoie le nombre : **5** *(La ville 3 possède 5 villes voisines)*
- La fonction **degre (0, R)** renvoie le nombre : **3** *(La ville 0 possède 3 villes voisines)*
- La fonction **degre (2, R)** renvoie le nombre : **4** *(La ville 2 possède 4 villes voisines)*

I. 4- Liste des degrés des villes

Q 4. Écrire la fonction **liste_degres (R)**, qui reçoit en paramètre la matrice symétrique **R** représentant un réseau routier. La fonction renvoie une liste **D** contenant des tuples. Chaque tuple de **D** est composé de deux éléments : une **ville** du réseau routier, et le **degré** de cette ville.

Exemple :

La fonction **liste_degres (R)** renvoie la liste **D** suivante :

D = [(0 , 3) , (1 , 3) , (2 , 4) , (3 , 5) , (4 , 5) , (5, 3) , (6 , 4) , (7 , 3)]

I. 5- Tri des villes

Q 5. a- Écrire la fonction **tri_degres (D)**, qui reçoit en paramètre la liste **D** des degrés des villes. La fonction trie les tuples de la liste **D** dans l'ordre décroissant des degrés des villes.

Exemple :

D = [(0 , 3) , (1 , 3) , (2 , 4) , (3 , 5) , (4 , 5) , (5, 3) , (6 , 4) , (7 , 3)]

Après l'appel de la fonction **tri_degres (D)**, on obtient la liste suivante :

[(3 , 5) , (4 , 5) , (2 , 4) , (6 , 4) , (0 , 3) , (5 , 3) , (1 , 3) , (7 , 3)]

Q 5. b- Écrire la fonction **tri_villes (R)**, qui reçoit en paramètre la matrice symétrique **R** représentant un réseau routier. La fonction renvoie la liste des villes triées dans l'ordre décroissant des degrés des villes.

Exemple :

La fonction **tri_villes (R)** renvoie la liste des villes triées dans l'ordre décroissant des degrés :

[3 , 4 , 2 , 6 , 0 , 5 , 1 , 7]

Partie II :

Coloration optimale des villes d'un réseau routier

Une **coloration** des villes du réseau routier est une affectation de couleurs à chaque ville, de façon à ce que deux villes voisines soient affectées par deux couleurs différentes. Le nombre minimum de couleurs nécessaires pour colorier un réseau routier est appelé : **nombre chromatique**.

La recherche du nombre chromatique est une question qui intervient dans beaucoup de problèmes concrets :

- ✓ L'établissement d'emplois du temps ;
- ✓ la gestion de ressources partagées ;
- ✓ la compilation dans l'utilisation efficace des registres ;
- ✓ ...

On cherche à construire une liste **C** qui contiendra les couleurs des villes. Ces couleurs seront représentés par des entiers strictement positifs : *chaque élément **C[k]** contiendra la couleur de la ville **k** du réseau routier.*

Pour construire la liste **C** des couleurs, on propose d'utiliser un algorithme, appelé : **algorithme de glouton**. C'est un algorithme couramment utilisé dans la résolution de ce genre de problèmes, afin d'obtenir des solutions optimales.

Principe de l'algorithme de glouton :

V est la liste des villes triées dans l'ordre décroissant des degrés des villes

C est une liste de même taille que **V**, initialisée par des **0**

Pour toute ville **k** de **V** :

C[k] ← première couleur non utilisée par les villes voisines à la ville **k**

Retourner la liste **C**

II. 6- Premier entier qui n'existe pas dans une liste d'entiers

Q 6- Écrire la fonction **premier_entier(L)**, qui reçoit en paramètre une liste **L** de nombres entiers positifs. La fonction renvoie le premier entier positif qui n'appartient pas à la liste **L**.

Exemple :

La fonction **premier_entier ([0 , 0 , 3 , 1 , 0 , 1 , 0 , 0])** renvoie le nombre : **2**

II. 7- Liste des couleurs des villes voisines à une ville

Q 7- Écrire la fonction **couleurs_voisines(k, C, R)**, qui reçoit en paramètres une ville **k** d'un réseau routier représenté par la matrice symétrique **R**, et la liste **C** des couleurs des villes du réseau. La fonction renvoie la liste des **C[i]** telle que **i** est une ville voisine à la ville **k**.

Exemples :

- La fonction **couleurs_voisines (4, [0, 0, 0, 1, 0, 0, 0, 0], R)** renvoie la liste : **[0, 0, 1, 0, 0]**
- La fonction **couleurs_voisines (2, [0, 0, 0, 1, 2, 0, 0, 0], R)** renvoie la liste: **[0, 0, 1, 2]**
- La fonction **couleurs_voisines (6, [0, 0, 3, 1, 2, 0, 0, 0], R)** renvoie la liste: **[1, 2, 0, 0]**
- La fonction **couleurs_voisines (0, [0, 0, 3, 1, 2, 0, 3, 0], R)** renvoie la liste: **[0, 3, 1]**

II. 8- Coloration des villes

Q 8- Écrire la fonction **couleurs_villes(R)**, qui reçoit en paramètre la matrice symétrique **R** représentant un réseau routier. La fonction renvoie la liste **C** des couleurs des villes, en utilisant le principe de glouton cité ci-dessus.

Exemple :

La fonction **couleurs_villes (R)** renvoie la liste des couleurs : **C = [2 , 1 , 3 , 1 , 2 , 1 , 3 , 2]**

Dans cette liste **C** :

- Les villes **0, 4** et **7** ont la même couleur : **2** ;
- Les villes **1, 3** et **5** ont la même couleur : **1** ;
- Les villes **2** et **6** ont la même couleur : **3**.

*NB : Dans cet exemple, le nombre chromatique est **3**.*