

DM 1

À rendre le 07/11/2024

Exercice 1

Dans cet exercice d'informatique commune, on se propose d'écrire des algorithmes dans le but de faire du calcul matriciel et plus particulièrement afin d'utiliser les matrices d'adjacence d'un graphe. Les algorithmes demandés doivent être écrits en langage **Python**. On sera très attentif à la rédaction et notamment à l'indentation du code. L'usage de toute librairie est **interdit**.

Notation

Les matrices sont carrées et représentées par des listes dont les éléments correspondent aux lignes de la matrice. Par exemple, la matrice $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ est représentée par la liste `[[1, 2], [3, 4]]`.

Dans la suite, pour définir la matrice d'adjacence $A \in \mathcal{M}_n(\mathbb{R})$ d'un graphe ayant n sommets, on numérote ses sommets de 0 à $n - 1$.

1. Écrire une fonction *produit* (A, B) prenant en arguments deux matrices carrées A et B de mêmes dimensions et qui renvoie AB le produit de la matrice A par la matrice B .
2. Écrire une fonction *orienté* (A) prenant en argument la matrice d'adjacence A d'un graphe et qui retourne *True* si le graphe est orienté et *False* sinon.
3. On admet que le nombre de chemins de longueur p reliant i et j dans un graphe de matrice d'adjacence A est égal au coefficient d'indice (i, j) de la matrice A^p .

Écrire une fonction *distance* (A, i, j) où A est la matrice d'adjacence d'un graphe et qui renvoie le nombre minimal d'arêtes que l'on doit parcourir pour atteindre le sommet j depuis le sommet i (on suppose qu'un tel chemin existe).

On considère deux tables : **CLIENTS** et **PARTENAIRES**. La première contient des informations sur les clients et la deuxième permet d'identifier qui sont les partenaires des clients.

La table **CLIENTS** contient les attributs suivants :

- **id** : identifiant d'un individu (entier), clé primaire ;
- **nom** (chaîne de caractères) ;
- **prenom** (chaîne de caractères) ;
- **ville** (chaîne de caractères) ;
- **email** (chaîne de caractères).

La table **PARTENAIRES** contient les attributs suivants :

- **id** : identifiant de suivi (entier), clé primaire ;
- **id_client** : identifiant du client représenté par l'attribut **id** dans la table **CLIENTS** (entier) ;

- **partenaire** : nom du partenaire (chaîne de caractères).
4. Écrire une requête SQL permettant d'extraire les identifiants de tous les clients provenant de la ville de "Toulouse".
 5. Écrire une requête SQL permettant d'extraire les emails de tous les clients ayant "SCEI" comme partenaire.
 6. (ajoutée)
Démontrer la propriété admise à la question 3.

Exercice 2

Un tensiomètre est utilisé dans le domaine hospitalier lors des visites régulières des infirmières dans les chambres des patients. Le tensiomètre doit permettre d'enregistrer les différentes constantes pour chaque patient. Après utilisation, le tensiomètre est connecté au réseau de l'hôpital et synchronise ses mesures avec la base de données des patients. La base de données est composée de plusieurs tables dont :

1. patients avec les attributs :

- **id** : identifiant associé à un patient, clé primaire
- **numero_secu** : le numéro de sécurité sociale du patient
- **nom** : le nom du patient
- **prenom** : le prénom (ou les prénoms) du patient
- **adresse1** : adresse du patient
- **adresse2** : complément d'adresse du patient
- **code_postal** : code postal de la ville du patient
- **ville** : ville du patient
- **date_de_naissance** : date de naissance du patient
- **telephone** : numéro de téléphone
- ...

2. mesures avec les attributs :

- **id** : identifiant associé à une mesure, clé primaire
- **type** : le type de mesure stockée : résultat d'analyse sanguine, pression artérielle, radio... (sous la forme d'une chaîne de caractères)
- **datetime** : la date et l'heure de l'examen sous la forme '2016-01-01 00 :00 :00'
- **pid** : identifiant du patient associé à la mesure
- **pdias** : pression diastolique en mmHg.
- **psyst** : pression systolique en mmHg.
- **pouls** : pouls en pulsations par minute.
- ...

Remarques : Les tables contiennent bien d'autres attributs, la base de données d'autres tables, seuls des éléments nécessaires à la suite ont été mis en place.

Si une valeur est stockée dans une variable, elle peut être utilisée dans une requête SQL.

Question 1 – Rappeler en quelques lignes l'utilité d'une clé primaire dans chacune des tables et dire si un autre attribut des tables aurait pu servir de clé primaire.

Pour le besoin du suivi médical des patients, les médecins doivent pouvoir avoir accès à l'évolution des constantes en fonction du temps lors d'une hospitalisation. A l'aide d'une interface graphique non étudiée ici, le médecin peut sélectionner les données à extraire entre deux dates qui seront stockées dans les variables `time1` et `time2` stockées dans le format de `datetime`. Il souhaite tracer les évolutions des pressions systolique, diastolique et le pouls entre ces deux dates.

Question 2 – Donner la requête SQL à envoyer au serveur de gestion de la base de données permettant de récupérer la valeur des champs `datetime`, `pdias`, `psyst` et `pouls` entre les deux dates `time1` et `time2` (stockées dans le format de `datetime`).

La requête précédente renvoie toutes les mesures sans distinction de type de mesures et sans que ce soit lié à un unique patient. Le médecin a sélectionné un patient avec son identifiant unique `id`, cette valeur est stockée dans la variable `id_patient`.

Question 3 – Donner la requête SQL à envoyer au serveur de gestion de la base de données permettant de récupérer la valeur des champs précédents entre les deux mêmes dates, pour le patient `id_patient` et pour un type de mesures '`tension`'.

A coté du graphique, le médecin souhaite avoir les valeurs minimale, maximale, moyenne et médiane de chacune des constantes.

Question 4 – Écrire une fonction `analyse(valeurs)` qui prend en argument une liste de valeurs et qui retourne le minimum, le maximum et la moyenne (sans utiliser ces fonctions prédéfinies en python).

Question 5 – Déterminer la complexité de votre fonction `analyse(valeurs)` en fonction de la taille de la liste de valeurs notée `n` dans le meilleur et le pire des cas. Vous calculerez pour cela le nombre de comparaisons et d'opérations (addition, soustraction, division, multiplication) réalisées.

Question 6 – A l'aide d'une méthode de tri de votre choix que vous explicitez, écrire une fonction `mediane(valeurs)` qui prend en argument une liste de valeurs et qui renvoie la valeur de la médiane.

Question 7 – Déterminer la complexité de votre fonction dans le meilleur et dans le pire des cas en comptant le nombre de comparaisons effectuées. Votre méthode est-elle efficace ? Justifier.

Question 8 – Le médecin souhaite obtenir les nom, prénom et le numéro de téléphone des patients qui ont eu (au moins ne fois) une pression systolique supérieure strictement à 160 mmHg, une pression diastolique supérieure strictement à 110 mmHg, avec un pouls compris strictement entre 100 et 150.

Ecrire la requête SQL à envoyer au serveur de gestion de la base de données permettant de répondre à la demande du médecin.

Question 9 – Le médecin souhaite obtenir un tableau contenant l'identifiant de chaque patient ainsi que le pouls minimum relevé pour ce patient.

Ecrire la requête SQL à envoyer au serveur de gestion de la base de données permettant de répondre à la demande du médecin.