Contrôle d'informatique – type CCINP
- e3a
MP
mardi 6 février 2023

Questions proches du cours

1. Écrire une fonction récursive, avec dictionnaire de mémoïsation initialisé par dico = {} avant la fonction, pour le calcul récursif de F_n, où

$$\begin{cases} F_0 & = 1 \\ F_1 & = 1 \\ F_{n+2} & = F_{n+1} + F_n \text{ pour } n \in \mathbb{N} \end{cases}$$

2. Une matrice est saisie comme une liste de listes. Par exemple, `A = [[1, 2], [3, 4]]`. Écrire une fonction prenant en paramètre d'entrée une matrice A carrée d'ordre n et qui renvoie `True` si A est triangulaire supérieure et `False` sinon.

Exercice 1

On dispose d'une base de données relationnelle `Vagues`.

La première table est `Bouee`. On se limite aux attributs suivants : le numéro d'identification `idBouee`, le nom du site `nomSite`, le nom de la mer ou de l'océan `localisation`, le type du capteur `typeCapteur` et la fréquence d'échantillonnage `frequence`.

`Bouee`

`idBouee`	`nomSite`	`localisation`	`typeCapteur`	`frequence`
831	Porquerolles	Mediterranee	Datawell non directionnelle	2.00
291	Les pierres noires	Mer d'iroise	Datawell directionnelle	1.28
...	...	...	...	...

La seconde table est `Campagne`. On se limite aux attributs suivants : le numéro d'identification `idCampagne`, le numéro d'identification de la bouée `idBouee`, la date de début `debutCampagne` et la date de fin `finCampagne`.

`Campagne`

`idCampagne`	`idBouee`	`debutCampagne`	`finCampagne`
08301	831	01/01/2010 00h00	15/01/2010 00h00
02911	291	15/10/2005 18h30	18/10/2005 08h00
...	...	...	...

La troisième table est `Tempete`. Les informations fournies relatives à un évènement « tempête » sont les suivantes :

— date de début et fin de tempête ;
— évolution de Hmax en fonction du temps ;
— le détail de certains paramètre non définis ici, obtenus au pic de tempête.

On se limite aux attributs suivants : le numéro d'identification de la tempête `idTempete`, le numéro d'identification de la bouéee `idBouee`, la date de début `debutTempete`, la date de fin `finTempete`, la valeur maximale de hauteur de vague `Hmax`.

Tempete

idTempete	idBouee	debutTempete	finTempete	Hmax
083010	831	07/01/2010 20h00	09/01/2010 15h30	5.3
029012	291	16/10/2005 08h30	18/10/2005 09h00	8.5
...	...	...	...	...

Le schéma de la base de donnée est donc : `Vagues=Bouee,Campagne,Tempete`.
Formuler les requêtes SQL permettant de répondre aux questions suivantes :

1. « Quels sont le numéro d'identification et le nom de site des bouées localisées en Méditerranée ? »
2. « Quel est le numéro d'identification des bouées où il n'y a pas eu de tempêtes ? »
3. « Pour chaque site, quelle est la hauteur maximale enregistrée lors d'une tempête ? »

Exercice 2

Les algorithmes demandés doivent être écrits en langage Python. On sera très attentif à la rédaction du code, notamment à l'indentation.

1. Donner la décomposition binaire (en base 2) de l'entier 21.
2. On considère la fonction `mystere` suivante.

```python
def mystere(n, b):
    """
    Donnees : n>0 un entier et b>0 un entier
    """
    t = [] # tableau vide
    while n > 0:
        c = n % b
        t.append(c)
        n = n // b
    return t
```

Pour $k \in \mathbb{N}^*$, on note c_k, t_k et n_k les valeurs prises par les variables `c, t, n` à la sortie de la k-ième itération de la boucle `while`. Recopier et compléter le tableau suivant, en ajoutant les colonnes nécessaires pour tracer entièrement l'exécution de `mystere(256, 10)` et préciser la valeur renvoyée après appel de `mystere(256, 10)` :

k	1	2	...
c_k			
t_k			
n_k			

3. Soit $n > 0$ un entier. On exécute `mystere(n,10)`. On pose $n_0 = n$.
   (a) Justifier la terminaison de la boucle `while`.
   (b) On note p le nombre d'itérations lors de l'exécution de `mystere(n,10)`. Justifier que pour tout $k \in [\![0, p]\!]$, on a $n_k \leqslant \frac{n}{10^k}$. En déduire une majoration de p en fonction de n.
4. En s'aidant du script de la fonction `mystere`, écrire une fonction `somme_chiffres` qui prend en argument un entier naturel et renvoie la somme de ses chiffres. Par exemple, `somme_chiffres(256)` devra renvoyer 13.
5. Écrire une version récursive de la fonction `somme_chiffres`, on la nommera `somme_rec`.

Exercice 3

1. Proposer une fonction Python `maxi` prenant en argument une liste d'entiers naturels `L` et renvoyant le maximum des entiers de cette liste.
   On n'utilisera pas de fonction spécifique de python déterminant ce maximum.

2. Écrire une fonction `ind` prenant en argument une liste d'entiers naturels `L` et renvoyant la liste des indices `[i_1,...,i_r]` avec `i_1<...<i_r` telle que pour tout $k \in [\![1, r]\!]$, `L[i_k]` soit non nul.

   Par exemple si `L = [0,1,3,0,7]`, alors `ind(L)` renvoie `[1,2,4]`.

3. Écrire une fonction `nb_oc` prenant comme argument une liste d'entiers naturels `L` et renvoyant la liste `T` de longueur `M = maxi(L)+1` où, pour tout $i \in [\![0, M]\!]$, `T[i]` est le nombre d'occurrences de l'entier `i` dans la liste `L`.

   Par exemple, si `L = [3,1,4,1,5]`, alors `T = [0,2,0,1,1,1]`.

   On pourra utiliser la fonction `maxi`.

4. Soit `L` une liste d'entiers naturels.

   (a) Déterminer le nombre de fois, noté `n`, où la liste `L` est parcourue lors de l'exécution de `nb_oc(L)`.

   (b) On veut que ce nombre `n` soit indépendant de `M = maxi(L)+1`.
   Si ce n'est pas le cas, modifier la fonction `nb_oc` afin de respecter cette condition.

5. Soit A une liste d'entiers naturels. On définit la suite de Robinson $(L_n)_{n\in\mathbb{N}}$ associée à la suite A par récurrence comme suit :

   — $L_0 = A$.
   — Si L_n est construite, alors :
     — on détermine $T_n = \texttt{nb\_oc}(L_n)$.
     — on détermine $I_n = \texttt{ind}(T_n)$.
     — si $I_n = [i_1, \ldots, i_r]$, alors $L_{n+1} = [T[i_r], i_r, \ldots, T[i_1], i_1]$.

   Par exemple si $A = [4, 4, 1, 2]$, alors :
   — $L_0 = [4, 4, 1, 2]$
   — $L_1 = [2, 4, 1, 2, 1, 1]$ (il y a deux « 4 », un « 2 » et un « 1 » dans la liste L_0)
   — $L_2 = [1, 4, 2, 2, 3, 1]$ (il y a un « 4 », deux « 2 » et trois « 1 » dans la liste L_1)

   (a) On donne $A = [2, 0, 4, 1, 3, 3, 2, 3, 1, 1]$. Déterminer L_3 et L_{2024}.

   (b) On donne $B = [2, 4, 1, 1, 1, 2]$. Si l'on suppose que $L_1 = B$, donner toutes les solutions possibles pour L_0.

   (c) On donne $C = [2, 4, 1, 0]$. Si l'on suppose que $L_1 = C$, donner toutes les solutions possibles pour L_0.

   (d) Proposer une fonction `rob(A,n)` qui prend en arguments une liste d'entiers naturels `A` et un entier naturel `n` et qui renvoie l'élément L_n de la suite de Robinson associée à `A`.
