PSI DS d'informatique N°1 12-01-2026

Exercice 1 : Base de données d'hôtel

Mise en situation et desciprtion de la base de données

On dispose d'une base de données d'un hôtel recensant les chambres, leur occupation, et les clients. La structure de cette base de données est données par la figure 1 :

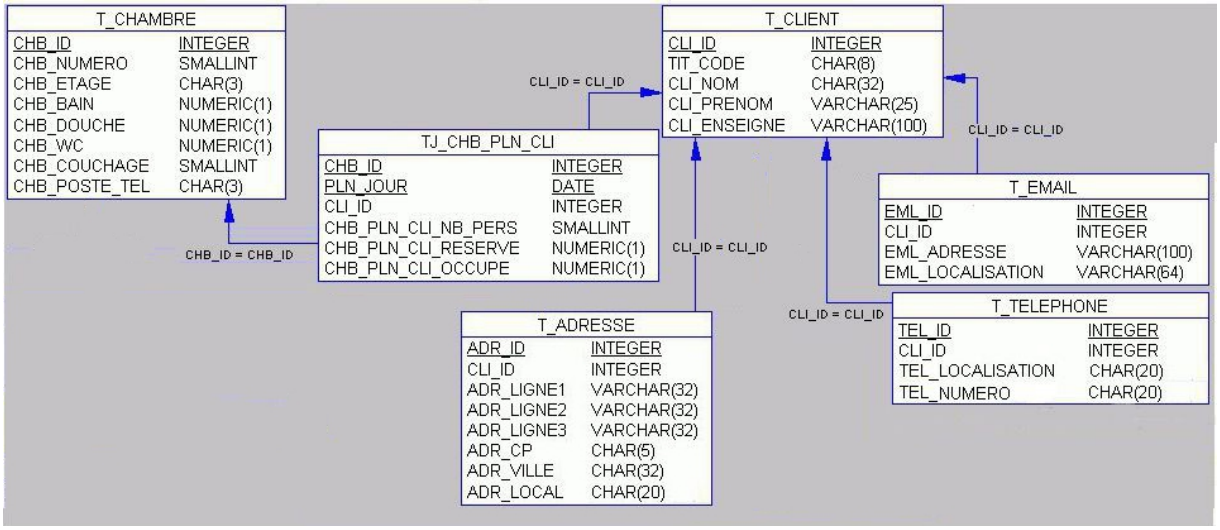

Figure 1 : *Structure de la base de donnée*

Les données des clients :

- « T_CLIENT » : Table listant les clients : Identifiants, noms, prénoms, civilités et entreprises.
- « T_ADRESSE » : Table listant les adresses postales des clients : Identifiants adresses, identifiants clients, localisations (3 lignes), Codes postaux, Villes et types d'adresses (« personnelle » ou « professionnelle »)
- « T_EMAIL » : Table listant les adresses électroniques des clients : Identifiants adresses, identifiants clients, adresses mail, types d'adresses (« personnelle » ou « professionnelle »)
- « T_TELEPHONE » : Table listant les numéros de téléphones des clients : Identifiants téléphones, identifiants clients, types de téléphones (« personnelle » ou « professionnelle »)

Les données des chambres :

- « T_CHAMBRE » : Tables listant les caractéristiques des chambres : Identifiants des chambres, numéros, étages, Baignoire (1 si oui 0 sinon), Douche (1 si oui 0 sinon), WC (1 si oui 0 sinon), Nombres de couchages, numéros du poste de la chambre.
- « TJ_TRF_CHB » : Table listant les tarifs par chambre et la date de début d'application du tarif (La date de fin d'application de ces tarifs étant la date postérieure suivante) : Identifiant des chambres, Dates de début d'application du tarif, tarifs.
- « T_TARIF » : Tables listant les taux de TVA et le tarif du petit déjeuner ainsi que la date de début d'application de ces tarifs (La date de fin d'application de ces tarifs étant la date postérieure suivante) :

Les données d'occupation et de réservation :

- « TJ_CHB_PLN_CLI » : Table permettant de faire la jonction entre les données des clients et les données des chambres. Cela permet de lister les occupations des chambres et leurs réservations par les clients : Identifiants des chambres, Jour de planification (occupation ou planification), Identifiants du client, Nombres de personnes dans la chambre, Réservations des chambres (1 si oui 0 sinon), Occupations de la chambre (1 si oui 0 sinon),

Travail demandé

Ecrire les requêtes permettant de :

Q1 - Lister les numéros des chambres de l'hôtel ayant au moins trois couchages avec leur nombre de couchages.

Q2 - Donner la capacité moyenne des chambres de l'hôtel.

Q3 - Lister les étages et leur nombre de chambres.

Q4 - Quel sont les étages qui ont une capacité moyenne des chambres supérieures ou égales à 2,5 ? Lister les étages et leur capacité moyenne des chambres.

Q5 - Lister les noms et prénoms des clients ayant une adresse personnelle à PARIS.

Q6 - Lister les noms et prénoms des clients qui ont donné une adresse personnelle et une adresse professionnelle.

Q7 - Donner la liste des personnes (Nom Prénom et numéro de téléphone) qui occupaient une chambre au premier étage de l'hôtel le 11 septembre 2025 ('2025-09-11')

Exercice 2 : Optimisation de la furtivité de la trajectoire d'un drône

Mise en situation

On souhaite calculer la trajectoire que doit prendre un drône militaire pour se déplacer d'un point de départ D vers un point d'arrivée A (Voir figure 2) en minimisant ses chances d'être détecté par l'ennemi.

Ce déplacement se fait dans une zone quadrillée par (p+1)×(q+1) carrés dont les arêtes ont pour longueur (et largeur) la distance : 2.a. Le point D est au centre du carré situé au Nord-Ouest de la zone et le point A est au centre du carré au Sud-Est de la zone. Donc dans un repère Ouest-Est ; Nord-Sud : (D, $\overrightarrow{x}$, $\overrightarrow{y}$) les points D et A ont respectivement pour coordonnées (0 ; 0) et (p×2.a ; q×2.a)

Modélisation du risque de détection

La topographie de la zone quadrillée, les obstacles obligeant le drône à prendre de l'altitude, et la présence des moyens de détection de l'ennemi permettent d'établir une cartographie du risque qu'a le drône d'être détecté dans chacun des carrés du quadrillage.

Le risque est modélisé par un entier n allant de 0 à 3. Sur la figure 2 cela est schématisé par quatre niveaux de gris différents :

Blanc : n = 0 → risque nul

Gris clair : n = 1 → risque faible

Gris foncé : n = 2 → risque moyen

Noir : n = 3 → risque élevé

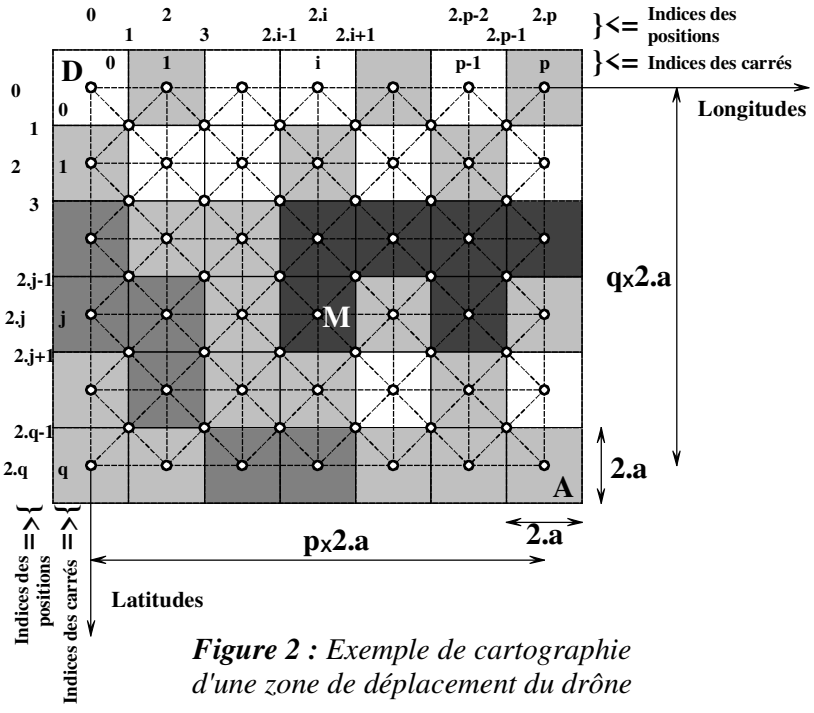

Figure 2 : *Exemple de cartographie d'une zone de déplacement du drône*

Cette cartographie est codée en python par une liste de listes contenant les entiers n. Chaque sous liste est une des colonnes de la zone. Par exemple, pour la zone de déplacement de la figure 2 on a la liste de listes **zone** ci-contre :

```
zone = [[0,1,2,2,1,1],
        [1,0,1,2,2,1],
        [0,0,1,1,1,2],
        [0,1,3,3,1,2],
        [1,0,3,1,0,1],
        [1,0,2,2,1,1],
        [1,0,2,1,0,1]]
```

Déplacements et positons du drône

Le déplacement de D à A se fait par des déplacements élémentaires entre des positions qui sont les centres ou les sommets des carrés (voir points blanc de la figure 2). Les déplacements élémentaires entre les différentes positions sont de 6 types différents (2 vers l'Est, 2 vers le Sud et 2 vers le Sud-Est) :

Type 1 : Du centre d'un carré au centre d'un carré voisin vers l'Est.
Type 2 : Du sommet d'un carré à un autre sommet voisin vers l'Est.
Type 3 : Du centre d'un carré au centre d'un carré voisin vers le Sud.
Type 4 : Du sommet d'un carré à un autre sommet voisin vers le Sud.
Type 5 : Du centre d'un carré à son sommet vers le Sud-Est
Type 6 : Du sommet d'un carré à son centre vers le Sud-Est

Chaque déplacement élémentaire est donc soit un déplacement vers l'Est d'une distance 2.a soit vers le Sud d'une distance 2.a soit vers le Sud-Est d'une distance $a.\sqrt{2} \approx 1{,}4.a$.

Chaque position (chaque point M) est définie par un couple d'entiers (i,j) tel que les coordonnées du point M sont : (i.a ; j.a) avec i et j des entiers pairs si le point M est le centre d'un carré ou i et j impairs si le point M est le sommet d'un carré. On a au total : $p \times q + (p-1) \times (q-1)$ positions différentes du drône.

Risque lié à un déplacement

On montre « facilement » que quelque soit la trajectoire empruntée pour aller de D à A il faut p + q déplacements élémentaires entre p + q + 1 positions M_k avec M_0 = D et M_{p+q} = A.

Le risque lié à la trajectoire prise pour aller du point de départ D au point d'arrivée A est la somme des risques liée à chaque déplacement élémentaire entre les point .M_k et M_{k+1}.

Hypothèses et données du codage

On suppose pour tout le problème que :

- ☞ Chaque carré de la zone à un coté de longueur égale à 2 c'est-à-dire que **a = 1**
- ☞ Au début du code python on a la ligne : **`import numpy as np`**

A- Construction du dictionnaire des risques des déplacements élémentaires

Chaque déplacement élémentaire comporte un risque de détection dont le calcul est le suivant :

Type 1 : Du centre du carré (i,j) au centre du carré (i+1,j) voisin vers l'Est : De la position (2.i,2.j) à la position (2.i+1,2.j) C'est la moyenne du risque dans le carré de (i,j) et de celle du carré (i+1,j) multipliée par la distance parcourue : 2.a.

Type 2 : Du sommet Sud-Ouest du carré (i,j) au Sud-Est vers l'Est : De la position (2.i-1,2.j+1) à la position (2.i+1,2.j+1) C'est la moyenne du risque dans le carré de (i,j) et de celle du carré (i,j+1) multipliée par la distance parcourue : 2.a.

Type 3 : Du centre du carré (i,j) au centre du carré (i,j+1) voisin vers le Sud : De la position (2.i,2.j) à la position (2.i,2.j+1) C'est la moyenne du risque dans le carré de (i,j) et de celle du carré (i,j+1) multipliée par la distance parcourue : 2.a.

Type 4 : Du sommet Nord-Est du carré (i,j) au Sud-Est vers le Sud : De la position (2.i+1,2.j-1) à la position (2.i+1,2.j+1) C'est la moyenne du risque dans le carré de (i,j) et de celle du carré (i+1,j) multipliée par la distance parcourue : 2.a.

Type 5 : Du centre du carré (i,j) à son sommet Sud-Est vers le Sud-Est : De la position (2.i,2.j) à la position (2.i+1,2.j+1) C'est le risque dans le carré de (i,j) multipliée par la distance parcourue : $a.\sqrt{2}$

Type 6 : Du sommet Nord-Ouest du carré (i,j) à son centre vers le Sud-Est : De la position (2.i-1,2.j-1) à la position (2.i,2.j) C'est le risque dans le carré de (i,j) multipliée par la distance parcourue : $a.\sqrt{2}$.

L'ensemble des risques sont stockés dans une liste de listes nommée « zone » par un entier de 0 à 3. Chaque sous liste est la liste d'une colonne de la zone quadrillée. Par exemple le risque du carré d'indice (i,j) (Longitude,Latitude) est donné par la valeur retournée par : zone[i][j].

Tous les risques de chaque déplacement élémentaire possible dans la zone seront stockés dans un dictionnaire **`DR_deplac`**. Les clefs sont le couple des couples de positions de départ et d'arrivée du déplacement élémentaire. Par exemple pour le déplacement de la position (k_d,ℓ_d) à la positon (k_a,ℓ_a) la clef est le couple $((k_d,\ell_d), (k_a,\ell_a))$. La valeur correspondant est le risque calculé comme indiqué ci-dessus. Par exemple : **`DR_deplac[((0,0),(2,0))]`** devra retourner **`1`** ou **`DR_deplac[((2,0),(3,1))]`** devra retourner **`1.4142135623730951`**

Q1 - On a implémenté la fonction **`def dic_risques_type1(zone:list) -> dict`** ci-dessous qui prend en argument une zone quadrillée « **`zone`** » et qui retourne un dictionnaire :

```
def dic_risques_type1(zone :list): -> dict
    Dico={}
    p,q=len(zone)-1,len(zone[0])-1
    for i in range(p):
        for j in range(q+1):
            Dico[((2*i,2*j),(2*i+2,2*j))]=zone[i][j]+zone[i+1][j]
    return Dico
```

A quoi correspond le dictionnaire retourné par cette fonction ?

Q2 - Compléter le code de la fonction **`def dic_risques_type2(zone:list) -> dict`** (lignes 4 à 6) qui prend en argument une zone quadrillée « **`zone`** » et qui retourne le dictionnaire de tous les déplacements élémentaires de type 2.

Q3 - Compléter le code de la fonction **`def dic_risques_type3(zone:list) -> dict`** (lignes 4 à 6) qui prend en argument une zone quadrillée « **`zone`** » et qui retourne le dictionnaire de tous les déplacements élémentaires de type 3.

Q4 - Compléter le code de la fonction **`def dic_risques_type5(zone:list) -> dict`** (lignes 4 à 6) qui prend en argument une zone quadrillée « **`zone`** » et qui retourne le dictionnaire de tous les déplacements élémentaires de type 5.

On suppose pour toute la suite du problème, qu'on à implémenté les deux fonctions **`def dic_risques_type4(zone:list)`** et **`def dic_risques_type6(zone:list)`** qui prennent en argument une zone quadrillée « **`zone`** » et qui retournent les dictionnaires de tous les déplacements élémentaires de type 4 et de type 6. On a également implémenté une fonction **`def dic_risques(zone:list) -> dict`** dont le code est :

```
def dic_risques(zone):
    DR_deplac=dic_risques_type1(zone)
    DR_deplac.update(dic_risques_type2(zone))
    DR_deplac.update(dic_risques_type3(zone))
    DR_deplac.update(dic_risques_type4(zone))
    DR_deplac.update(dic_risques_type5(zone))
    DR_deplac.update(dic_risques_type6(zone))
    return DR_deplac
```

Ainsi cette fonction permet d'obtenir par la commande **`dic_risques(zone)`** le dictionnaire des risques de tous les déplacements élémentaires de l'ensemble de la zone.

B- Calcul du risque lié à une trajectoire

La trajectoire décrite pour aller du point de départ D au point d'arrivée A est donnée par une liste de chaines de caractère donnant la direction prise pour chaque déplacement élémentaire : `'S'` pour aller vers l'Est, `'S'` pour aller vers le Sud et `'SE'` pour aller vers le Sud-Est.

Q5 - Tracer sur le document réponse la trajectoire liée à la liste `['E', 'E', 'E', 'E', 'E', 'SE', 'SE', 'S', 'S', 'S', 'S']` puis calculer le risque lié à cette trajectoire.

Q6 - Implémenter la fonction `calcul_risque_trajectoire(L_directions :list, DR_deplac :dict)->float` qui prend en argument la liste `L_directions` d'une trajectoire du point D (position (i,j) = (0,0)) au point A, le dictionnaire `DR_deplac` des risques de tous les déplacements élémentaires et qui retourne le flottant donnant le risque lié à cette trajectoire.

C- Première stratégie

Une stratégie pour trouver la trajectoire optimale (minimisant le risque de détection du drône) est de calculer les risques liés à toutes les trajectoires possibles pour aller de D à A et de ne retenir que celle dont le risque est minimale.

Q7 - Quelle serait la complexité d'un algorithme utilisant cette stratégie ? Pourquoi va-t-on exclure cette stratégie algorithmique ?

La première stratégie pour minimiser les risques d'un déplacement de départ D à A sera donc la suivante : On part du point D (position (0,0)) pour arriver au point A (position (2.p,2.q). A chaque fois qu'on est sur une position (i,j) nous allons aller en direction de l'Est, du Sud ou du Sud-Est. Le choix d'une de ces trois directions se fera de la manière suivante :

On peut aller vers l'Est tant que $i \leq 2.p-2$, on peut aller vers le Sud tant que $j \leq 2.q-2$, on peut aller vers le Sud-Est tant que $i \leq 2.p-1$ et $j \leq 2.q-1$. On choisit parmi les directions possibles la direction donnant le déplacement élémentaire dont le risque est minimal.

Q8 - De quel type d'algorithme relève cette stratégie ?

Q9 - Tracer sur le document réponse la trajectoire du drône obtenue avec cette stratégie. Donner également le risque total résultant de votre trajectoire (utiliser $\sqrt{2} \approx 1{,}4$). Remarque : Il peut y avoir plusieurs réponses possibles, une seule est demandée.

Q10 - Implémenter une fonction `direction_moins_risquee(DR_deplac:dict, i:int,j:int,p:int,q:int)->str` qui prend en argument un dictionnaire `DR_deplac` des risques de tous les déplacements élémentaires, une position donnée par les entiers `i` et `j`, la dimension de la zone ((2.p+1)×(2.q+1)) donnée par les entiers `p` et `q`, et qui retourne la direction (`'E'`,`'S'` ou `'SE'`) correspondant au déplacement élémentaire dont le risque est minimal.

Q11 - On donne sur le document réponse les premières lignes de code de la fonction pour déterminer la trajectoire avec cette stratégie. `strategie1(zone:list)->tuple` qui prend en argument la liste `zone` correspondant à une cartographie des risques et qui retourne un couple `(risque_trajectoire,trajectoire)` où `trajectoire` est la trajectoire obtenue par la stratégie décrite ci-dessus et `risque_trajectoire` est le risque lié à cette trajectoire. Compléter le code de cette fonction.

Q12 - Cette stratégie est-elle optimale ? Justifier la réponse.

D- Deuxième stratégie

La première stratégie n'étant pas optimale on utilise une 2nd stratégie qui elle donnera une trajectoire optimale. Le principe est qu'en partant du point D pour arriver au point M de coordonnées (i,j) avec une trajectoire optimale, il faut (si M≠D) prendre la trajectoire qui donne le risque minimal parmi les trois trajectoires suivantes :

- ☞ Trajectoire optimale du point D au point M_O situé immédiatement à l'Ouest de M (quand celui-ci existe) dont les coordonnées sont (i–2,j) plus un déplacement vers l'Est.
- ☞ Trajectoire optimale du point D au point M_N situé immédiatement au Nord de M (quand celui-ci existe) dont les coordonnées sont (i,j–2) plus un déplacement vers le Sud.
- ☞ Trajectoire optimale du point D au point M_{NO} situé immédiatement au Nord-Ouest de M (quand celui-ci existe) dont les coordonnées sont (i–1,j–1) plus un déplacement vers le Sud-Est.

Ce type d'algorithme répond à une équation dite équation de Bellman.

On note $R_{opti}^{i,j}$ le risque lié à la trajectoire optimale de D à M, $R_{opti}^{i-2,j}$ le risque lié à la trajectoire optimale de D à M_O (lorsque M_O existe), $R_{opti}^{i,j-2}$ le risque lié à la trajectoire optimale de D à M_N (lorsque M_N existe), $R_{opti}^{i-1,j-1}$ le risque lié à la trajectoire optimale de D à M_{NO} (lorsque M_{NO} existe).

On note également $R_{déplacement}^{k_d,\ell_d \to k_a,\ell_a}$ le risque lié au déplacement du point (k_d,ℓ_d) au point (k_a,ℓ_a).

L'équation de Bellman de l'algorithme s'écrit alors :

$$R_{opti}^{i,j} = \text{mini}\left(R_{opti}^{i-2,j} + R_{déplacement}^{i-2,j \to i,j} , R_{opti}^{i,j-2} + R_{déplacement}^{i,j-2 \to i,j} , R_{opti}^{i-1,j-1} + R_{déplacement}^{i-1,j-1 \to i,j} \right)$$

La trajectoire optimale sur une zone de dimensions (p+1)×(q+1) pour un déplacement de D de coordonnées (0,0) au point A de coordonnées (2.p,2.q) est donc la trajectoire donnant le risque $R_{opti}^{2.p,2.q}$.

Q13 - De quel type d'algorithme relève cette stratégie ?

On donne le code Python de la fonction suivante qui prend en argument une liste de listes **zone** correspondant à la cartographie d'une zone de déplacement d'un drône :

```
1   def strategie2(zone):
2       DR_deplac=dic_risques(zone)
3       p,q=len(zone)-1,len(zone[0])-1
4       def traj_opt(i,j):
5           if (i,j)==(0,0):
6               return []
7           Ropt_O,Ropt_N,Ropt_NO=np.inf,np.inf,np.inf
8           if i>=2:
9               Topt_O=traj_opt(i-2,j)+['E']
10              Ropt_O=calcul_risque_trajectoire(Topt_O,DR_deplac)
11          if j>=2:
12              Topt_N=traj_opt(i,j-2)+['S']
13              Ropt_N=calcul_risque_trajectoire(Topt_N,DR_deplac)
14          if i>=1 and j>=1:
15              Topt_NO=traj_opt(i-1,j-1)+['SE']
16              Ropt_NO=calcul_risque_trajectoire(Topt_NO,DR_deplac)
17          Ropt=min(Ropt_O,Ropt_N,Ropt_NO)
18          if Ropt==Ropt_O:
19              Topt=Topt_O
20          if Ropt==Ropt_N:
21              Topt=Topt_N
22          if Ropt==Ropt_NO:
23              Topt=Topt_NO
24          return Topt
25      Topt_finale=traj_opt(2*p,2*q)
26      Ropt_final=calcul_risque_trajectoire(Topt_finale,DR_deplac)
27      return Ropt_final,Topt_finale
```

Q14 - Que retourne cette fonction ? Donner le type des éléments retournés.

Q15 - Quelles sont les lignes du code qui reprennent l'équation de Bellman de l'algorithme.

Q16 - De quel type est la sous fonction **`traj_opt(i,j)`** ?

Q17 - L'exécution de cette fonction pour une zone de dimensions 7×7 donne le résultat attendu quasi instantanément. Pour une dimension 8×8 elle donne le résultat avec un temps d'exécution plus long. Pour une dimension 9×9 ou 10×10 avec un temps excessivement long voir quasi infiniment long. Qu'est-ce qui justifie cela. Justifier votre réponse à partir du code de cette fonction donné ci-dessus.

Q18 - Implémenter une fonction **`strategie2_memoisation(zone)`** similaire à la fonction **`strategie2(zone)`** mais qui a une complexité linaire. Vous utiliserez pour cela un dictionnaire **`D_traj_opt`** dont les clefs sont les coordonnées des positions et les valeurs les trajectoires optimales pour rejoindre ces positions à partir du point D. La première clef de ce dictionnaire est **`(0,0)`** et la valeur correspondante **`[]`**. Soit : **`D_traj_opt[(0,0)]`** → **`[]`**.