Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

Exercice 1 - Algorithmes de tri

Partie A : Manipulation de listes

1. Le code Python affiche 8, étant le nombre d'éléments dans la liste puis il affiche la liste `[8, 7, 18, 16, 12, 9, 17, 3]`.

2. Pour afficher les éléments de la liste de l'indice 2 à 4 de la liste `notes`, on peut utiliser les codes :

Solution 1

```
print(notes[2:5])
```

Solution 2

```
for i in range(2, 5):
    print(notes[i])
```

Partie B : Tri par insertion

1.

```
while i >= 0 and liste[i] > element_a_inserer :
      liste[i+1] = liste[i]
```

2. Après une itération de la boucle `for`, la liste est `[7, 8, 18, 14, 12, 9, 17, 3]`.

3. Après trois itérations de la boucles `for`, la liste est `[7, 8, 14, 18, 12, 9, 17, 3]`.

Partie C : Tri Fusion

1. C'est un algorithme récursif car il s'appelle lui-même à la troisième étape.

2. Étape 1 : Comparer les cartes visibles (la première de chaque tas).
   Étape 2 : Placer la carte avec la valeur la plus petite dans la main.
   Étape 3 : Recommencer jusqu'à l'épuisement des tas.

3.

```
tri_fusion(liste, i_debut, i_partage)
tri_fusion(liste, i_partage+1, i_fin)
fusionner(liste, i_debut, i_partage, i_fin)
```

4. La première ligne du code de la question 3 sert à importer depuis le module `math` la fonction `floor`.
La fonction **`floor()`** renvoie la partie entière d'un nombre.

Partie D : Comparaison du tri par insertion et du tri fusion

1. C'est le tri fusion qui a été utilisé car à chaque étape on fusionne deux tas triés.

2. Le tri fusion a une complexité en $O(n \log_2 n)$ et le tri par insertion à une complexité en $O(n^2)$.

3. Le tri fusion divise la liste à trier en deux à chaque itération sur le principe « diviser pour régner ».
Le tri par insertion utilise deux boucles imbriquées parcourant la liste à trier.

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

Exercice 2 - Base de données d'une plateforme de vente en ligne

Partie A : Modèle relationnel

1. La clé primaire de la relation `Clients` est `IdClient`. La clé primaire de la relation `Articles` est `IdArticle`.

2. Le domaine de l'attribut `Email` est VARCHAR(50), c'est-à-dire une chaîne de caractère de longueur variable avec au maximum 50 caractères. Le domaine de l'attribut `Quantite` est `INT`, c'est-à-dire un entier.

3.

```
CREATE TABLE Commandes (
IdCmd  INT PRIMARY KEY,
IdClient INT,
Date DATE,
AdresseLivraison VARCHAR(90),
PaiementValide BOOLEAN,
LivraisonFaite BOOLEAN,
FOREIGN KEY(IdClient) REFERENCES Clients(IdClient)
);
```

Partie B : Site web

1. La méthode POST permet de transmettre les informations dans le corps de la requête HTTP, contrairement à la méthode GET qui les transmet via l'URL. Ainsi il est possible de transmettre une quantité importante de données.
(De plus, avec la méthode POST, nous ne sommes pas limités aux caractères ASCII.)

2. Le protocole HTTPS est chiffré de bout en bout ce qui est essentiel pour que les transactions bancaires soient confidentielles.

3. La vérification des informations saisies permet d'assurer la cohérence des données en vue de leur traitement ultérieur.

Partie C : Requêtes SQL

1.

```
SELECT IdArticle, Libelle FROM Articles WHERE PrixEnCentimes < 1500 ;
```

Remarque : Le domaine de l'attribut `PrixEnCentimes` est `INT`, c'est-à-dire un entier donc la valeur 1500 correspond bien à 15,00€.

2. La requête sélectionne les identifiants et les courriels des clients, ainsi que les identifiants et les adresses de livraison de leurs commandes pour des paiements qui n'ont pas été validés.

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

3.

```
SELECT Articles.Libelle FROM Articles
JOIN ArticlesCommande ON Articles.IDArticle = ArticlesCommande.IdArticle
JOIN Commandes ON ArticlesCommande.IdCmd = Commandes.IdCmd
WHERE Commandes.IdCmd = 1345 ;
```

4.

```
INSERT INTO Articles
VALUES (NULL, 'Imperméable', 'Cet imperméable se replie en forme de pochette.', 999) ;
```

Partie D : Adaptation du modèle relationnel

1. Dans la relation `Articles`, on ajoute l'attribut `stock` avec comme domaine `INT`.
Dans la relation Client, on ajoute l'attribut `AdresseLivraison` avec comme domaine `VARCHAR(90)`.

2. Il faut soustraire la quantité commandée au stock et pas seulement 1.

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

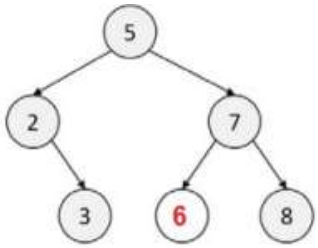
Exercice 3 - Arbre binaire de recherche

Partie A : Étude d'un exemple

1. Le nœud racine a pour valeur 5 et a pour fils les nœuds 2 et 7.

2. La branche qui a pour feuille la valeur 3 passe par les nœuds 5 et 2.

3.



Partie B : Implémentation en Python

1. La fonction `__init__` permet d'instancier un objet de cette classe en définissant ses attributs.

2. Si on essaye d'ajouter un élément déjà présent dans l'arbre, il va être ignoré par le programme.

3.

```
arbre = ABR(5)
arbre.insererElement(2)
arbre.insererElement(3)
arbre.insererElement(7)
arbre.insererElement(8)
```

Partie C : Tri par arbre binaire de recherche

1. Le parcours infixe permet de visiter les valeurs d'un arbre dans l'ordre croissant.

2. La complexité de la méthode de tri d'un arbre binaire de recherche est en $O(n \log_2 n)$ comme pour le tri fusion (méthode « diviser pour régner »), alors que pour le tri par insertion ou sélection la complexité est en $O(n^2)$ avec une double boucle imbriquée parcourant la liste à trier.

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

Exercice 4 - Au cœur des machines

Partie A : Routage dans un réseau informatique

1. Pour transmettre de grandes quantités de données, le protocole TCP-IP prévoit leur découpage en paquets de taille standardisée et encapsulés par des informations telles que l'adresse IP de l'expéditeur, du destinataire et le numéro du paquet. Ces paquets sont appelés des datagrammes et sont routés (commutés) de proche en proche par chaque routeur à l'aide des données stockées dans l'encapsulation des paquets et des tables de routage propres à chaque routeur.

2. a. Les sommets représentent les routeurs et les arêtes représentent les connexions entre les routeurs.

2. b. Le protocole RIP utilise le nombre de sauts (hops) entre routeurs comme métrique.

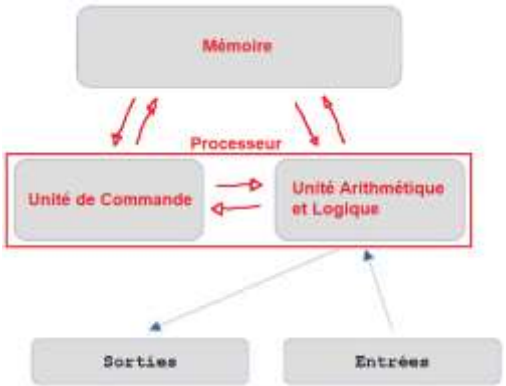
Partie B : Système d'exploitation

1. Une situation d'interblocage se produit lorsque deux processus se bloquent mutuellement soit parce qu'ils monopolisent chacun une ressource que l'autre attend pour poursuivre son exécution, soit parce qu'ils attendent un résultat l'un de l'autre.

2. Pour éviter une situation d'interblocage, il y a l'utilisation de mutex ou l'algorithme du Banquier.

Partie C : Architectures matérielles

1.

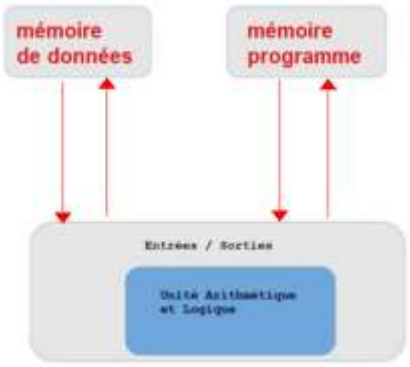


2. Le CP (compteur de programme) est un registre du processeur qui contient l'adresse mémoire de l'instruction prochainement exécutée ou en cours d'exécution. Quand l'exécution est terminée, il est incrémenté avec l'adresse mémoire de la prochaine instruction.
Le RI (registre d'instruction) est dans le processeur et contient l'instruction en cours de décodage ou d'exécution.

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

3. Architecture de Harvard :



4. La mémoire morte est non volatile (ne s'efface pas hors tension). Elle contient les informations nécessaires au fonctionnement d'un matériel et sont ainsi inscrite en « dur » par le fabricant. On peut la lire mais pas y écrire.
La mémoire vive est volatile (qui s'efface hors tension) et permet de stocker les données en cours d'utilisation pour en accélérer l'obtention par le processeur. On peut y écrire et lire.

Partie D : Système sur puce

1. Avoir plusieurs processeurs permet d'effectuer plusieurs calculs simultanément et ainsi augmenter la vitesse des calculs.

2. Les systèmes sur puces rassemblent plusieurs composants différents (processeur, mémoire, entrée/sortie, …) ayant possiblement des vitesses de transmission différentes.

3. Les systèmes sur puces permettent une plus grande intégration grâce à leur faible encombrement. La proximité des composants permet une plus grande rapidité de traitement et une consommation énergétique réduite. Enfin l'automatisation de leur production permet d'obtenir des coûts réduits.

4. Les inconvénients des systèmes sur puces sont :

- **un coût de recherche et développement** plus important,
- **leur obsolescence** du fait de l'impossible évolutivité. En effet les composants intégrés ne peuvent être séparés et remplacés individuellement,
- **la faible durée de vie**. La durée de vie du SoC correspond à celle de son composant le plus fragile. Plus y a de composants intégrés, plus il y a de chance qu'un composant soit défaillant rapidement et condamne le SoC tout entier.

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

Exercice 5 - Vidéos à la demande

Partie A : Modèle relationnel

1. Les clés primaires des relations `Films` et `Abonnes` sont respectivement `idFilm` et `idAbonne`.

2. Le domaine des attributs `idFilm` et `Description` sont respectivement `INT` (un nombre entier) et `VARCHAR(150)` (une chaîne de maximum 150 caractères).

3. Pour la relation `ComptesAbonnes`, la clé primaire est `idCpt` et la clé étrangère est `idAbonne`.

4. Il faudrait créer deux nouvelles relations `Acteurs` et `Roles` avec comme schéma relationnel :
`Acteurs(IdActeur, Nom, Prenom)`
`Roles(#IdActeur, #IdFilm)`

Pour la relation `Acteurs`, la clé primaire est `IdActeur`. Le domaine des attributs `Nom` et `Prenom` peut être `VARCHAR(30)`.
Pour la relation `Roles`, la clé primaire est le couple d'attributs `IdActeur` et `IdFilm`.

Il est impossible d'associer directement les relations `Films` et `Acteurs` car un même acteur peut jouer dans plusieurs films et un même film peut avoir plusieurs acteurs.

5. Il faut ajouter des attributs :

- `Date_Naissance` de domaine DATE dans la relation `Clients`
- `Age_Minimum` de domaine `INT` dans la relation `films`

Partie B : Requêtes SQL

1.

```sql
SELECT IdCpt, Pseudo FROM ComptesAbonnes WHERE IdAbonne = 237 ;
```

2. Cette requêtes SQL calcule une moyenne des notes attribuées au film avec l'identifiant 1542.

3. Elle sélectionne les identifiants, les titres et les notes attribuées à des films par le compte d'identifiant 508 en les triant par ordre décroissant des notes attribuées.

4.

```sql
UPDATE ComptesAbonnes SET pseudo = 'Champion' WHERE IdAbonne = 508 ;
```

Correction

NSI - 2021 Polynésie (21-NSIJ2PO1)

Partie C : Conseils de films

1. Cette fonction calcule la moyenne des écarts de notes attribuées à chaque film entre deux comptes.

2.

```
def conseilsFilms(IdCpt):
    conseils = []
    liste_films = podiumCompte(IdCpt)
    liste_spectateurs = spectateurs(liste_films)
    for spect in liste_spectateurs:
        if distance(IdCpt, spect, liste_films) < 10:
            conseils = podiumCompte(spect)[:3]
    return conseils
```