

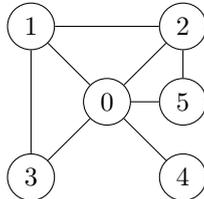
# Centres étrangers - juin 2024 - sujet 1 (corrigé)

## Exercice 1 (Programmation Python, programmation dynamique, graphes et réseaux)

1. On a la définition suivante :

```
voisins = [[1, 2, 3, 4],  
           [0, 2, 3],  
           [0, 1],  
           [0, 1],  
           [0]]
```

2. On a le graphe suivant :



3. On a la nouvelle définition suivante :

```
voisins = [[1, 2, 3, 4, 5],  
           [0, 2, 3],  
           [0, 1, 5],  
           [0, 1],  
           [0],  
           [0, 2]]
```

4. La fonction suivante convient :

```
def voisin_alea (voisins, s):  
    return voisins[s][random.randrange(len(voisins[s]))]
```

5. La fonction `marche_alea` est récursive car elle s'appelle elle-même.

6. Cette fonction permet de parcourir une chaîne dans le graphe `voisins`, cette chaîne commençant au sommet `i` et comportant `n` sommets. La fonction renvoie le dernier sommet de cette chaîne.

7. La fonction suivante convient :

```
def simule (voisins, i, n_tests, n_pas):  
    results = [0] * len (voisins)  
    while n_tests > 0:  
        j = marche_alea(voisins, i, n_pas)  
        results[j] = results[j] + 1  
        n_tests = n_tests - 1  
    return [s/n_tests for s in results]
```

8. Il s'agit de l'ordinateur 0 car dans 32,8% des cas, le virus termine sa course sur cet ordinateur.

9. Il suffit d'effectuer un parcours en largeur du graphe et de compter le nombre d'itérations nécessaires pour visiter tous les nœuds du graphe.

**Exercice 2 (Réseaux et protocoles de routage)****Partie A : adresses IP**

1. Il s'agit du masque  $255.255.0.0$
2. D'après la question A-1, l'adresse du réseau L2 est  $172.16.0.0$
3. D'après la question A-1, l'adresse de diffusion du réseau L2 est  $172.16.255.255$
4. Puisque les deux derniers octets sont disponibles, il y a donc  $256^2 - 2$  machines qui peuvent être connectées au réseau L2 au maximum.

**Partie B : protocoles de routage**

5. Un chemin possible : S1 - A - H - D - S2
6. Deux chemins possibles : S1 - A - H - C - D - S2 ou S1 - A - B - C - D - S2
7. \* Pour le premier chemin A-H-C-D, on a la modification suivante pour le routeur H :

Routeur	Réseau destinataire	Passerelle	Interface
H	L2	53.10.10.10	53.10.10.9

8. \* Pour le premier chemin A-B-C-D, on a la modification suivante pour le routeur A :

Routeur	Réseau destinataire	Passerelle	Interface
A	L2	193.55.24.1	193.55.24.2

9. On a les calculs de coûts suivants :
  - \*  $100 \text{ Mbit/s} = 10^8 \text{ bit/s}$ , soit un coût de 10
  - \*  $1 \text{ Gbit/s} = 10^9 \text{ bit/s}$ , soit un coût de 1
  - \*  $10 \text{ Gbit/s} = 10^{10} \text{ bit/s}$ , soit un coût de 0,1
10. On a le chemin suivant : S1 - A - G - F - E - D - S2 avec un coût de  $1 + 0,1 + 0,1 + 0,1 = 1,3$
11. On a le chemin suivant : S1 - A - H - F - E - D - S2 avec un coût de  $1 + 1 + 0,1 + 0,1 = 2,2$

**Exercice 3 (POO, bases de données et SQL)****Partie A**

1. Deux avantages :

- \* les bases de données relationnelles permettent de gérer la lecture, l'écriture ou la modification des informations contenues dans une base de données (plus difficile avec un fichier texte);
- \* les bases de données relationnelles permettent de gérer les autorisations d'accès.

2. Pour être utilisé comme clé primaire, un attribut doit avoir une valeur unique pour chaque entrée

3. `id_client` et `id_emplacement` sont des clés étrangères.

- \* `id_client` permet de relier (jointure) la table `Reservation` et la table `Client`;
- \* `id_emplacement` permet de relier (jointure) la table `Reservation` et la table `Emplacement`.

4. On a le schéma relationnel suivant :

`Emplacement(id_emplacement, nom, localisation, tarif_journalier)`

5. On a le tableau suivant :

1	myrtille	A4
4	mandarine	B1
6	melon	A2

6. La requête suivante convient :

```
SELECT nom, prenom FROM Client WHERE ville = 'Strasbourg'
```

7. La requête suivante convient :

```
INSERT INTO Client VALUES (42, 'CODD', 'Edgar', '28 rue des Capucines', 'Lyon', 'France', '0555555555')
```

8. La requête suivante convient :

```
SELECT Client.nom, Client.prenom, Reservation.nombre_personne,
Reservation.date_arrivee, Reservation.date_depart, Emplacement.tarif_journalier
FROM Reservation
JOIN Client ON Client.id_client = Reservation.id_client
JOIN Emplacement ON Emplacement.id_emplacement = Reservation.id_emplacement
WHERE Reservation.id_reservation = 18
```

**Partie B**

9. `self` correspond à l'instance courante, on passe cette instance courante en paramètres aux méthodes de la classe.

10. L'instruction suivante convient :

```
client01 = Client('CODD', 'Edgar', '28 rue des Capucines', 'Lyon', 'France', '0555555555')
```

11. Il faut saisir l'instruction suivante :

```
reservation.nb_jours() * emplacement.tarif_journalier +
2.20 * reservation.nb_jours() * reservation.nombre_personne
```

12. La variable `annee` est de type chaîne de caractères (`str`). A ligne 26, on cherche à comparer la variable `annee` avec les entiers 2018 et 2024, ce qui est impossible, puisque l'on ne peut pas comparer un entier et une chaîne de caractères. D'où l'erreur.

13. On peut modifier, par exemple, la ligne 26 comme suit :

```
2018 <= int(annee) <= 2024
```

14. On complète les lignes 29 et 31 comme suit :

\* Ligne 29 :

```
if mois not in calendrier:
    return False
```

\* Ligne 31 :

```
if len(numero) != 3 or not que_des_chiffres(numero):
    return False
```