

DS

**ENTRAÎNEMENT AU
BACCALAURÉAT GÉNÉRAL**

ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

NUMÉRIQUE ET SCIENCES INFORMATIQUES

Mercredi 14 décembre 2022

Durée de l'épreuve : **2 heures**

L'usage de la calculatrice n'est pas autorisé.

Le candidat traite les 3 exercices proposés

Chaque exercice est noté sur 4 points.

EXERCICE 1 (4 points)

Cet exercice porte sur les arbres binaires de recherche, la programmation orientée objet et la récursivité.

Dans cet exercice, la taille d'un arbre est le nombre de nœuds qu'il contient. Sa hauteur est le nombre de nœuds du plus long chemin qui joint le nœud racine à l'une des feuilles (nœuds sans sous-arbres). On convient que la hauteur d'un arbre ne contenant qu'un nœud vaut 1 et la hauteur de l'arbre vide vaut 0.

1. On considère l'arbre binaire représenté ci-dessous:

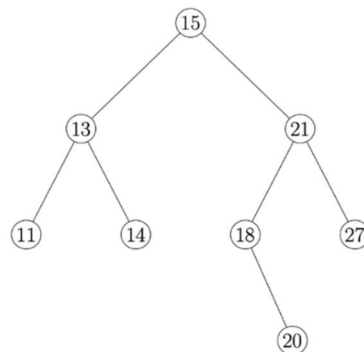


Figure 1

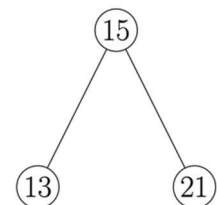
- Donner la taille de cet arbre.
- Donner la hauteur de cet arbre.
- Représenter sur la copie le sous-arbre droit du nœud de valeur 15.
- Justifier que l'arbre de la figure 1 est un arbre binaire de recherche.
- On insère la valeur 17 dans l'arbre de la figure 1 de telle sorte que 17 soit une nouvelle feuille de l'arbre et que le nouvel arbre obtenu soit encore un arbre binaire de recherche. Représenter sur la copie ce nouvel arbre.

2. On considère la classe `Noeud` définie de la façon suivante en Python :

```

1 class Noeud:
2     def __init__(self, g, v, d):
3         self.gauche = g
4         self.valeur = v
5         self.droit = d
  
```

- Parmi les trois instructions **(A)**, **(B)** et **(C)** suivantes, écrire sur la copie la lettre correspondant à celle qui construit et stocke dans la variable `abr` l'arbre représenté ci-contre.



- `abr=Noeud(Noeud(Noeud(None, 13, None), 15, None), 21, None)`
- `abr=Noeud(None, 13, Noeud(Noeud(None, 15, None), 21, None))`
- `abr=Noeud(Noeud(None, 13, None), 15, Noeud(None, 21, None))`

- b.** Recopier et compléter la ligne 7 du code de la fonction `ins` ci-dessous qui prend en paramètres une valeur `v` et un arbre binaire de recherche `abr` et qui renvoie l'arbre obtenu suite à l'insertion de la valeur `v` dans l'arbre `abr`. Les lignes 8 et 9 permettent de ne pas insérer la valeur `v` si celle-ci est déjà présente dans `abr`.

```

1 def ins(v, abr):
2     if abr is None:
3         return Noeud(None, v, None)
4     if v > abr.valeur:
5         return Noeud(abr.gauche, abr.valeur, ins(v, abr.droit))
6     elif v < abr.valeur:
7         return .....
8     else:
9         return abr

```

- 3.** La fonction `nb_sup` prend en paramètres une valeur `v` et un arbre binaire de recherche `abr` et renvoie le nombre de valeurs supérieures ou égales à la valeur `v` dans l'arbre `abr`.

Le code de cette fonction `nb_sup` est donné ci-dessous :

```

1 def nb_sup(v, abr):
2     if abr is None:
3         return 0
4     else:
5         if abr.valeur >= v:
6             return 1+nb_sup(v, abr.gauche)+nb_sup(v, abr.droit)
7         else:
8             return nb_sup(v, abr.gauche)+nb_sup(v, abr.droit)

```

- a.** On exécute l'instruction `nb_sup(16, abr)` dans laquelle `abr` est l'arbre initial de la figure 1. Déterminer le nombre d'appels à la fonction `nb_sup`.
- b.** L'arbre passé en paramètre étant un arbre binaire de recherche, on peut améliorer la fonction `nb_sup` précédente afin de réduire ce nombre d'appels. Écrire sur la copie le code modifié de cette fonction.

EXERCICE 2 (4 points)

Cet exercice porte sur les bases de données.

On pourra utiliser les mots clés SQL suivants : **SELECT**, **FROM**, **WHERE**, **JOIN**, **ON**, **INSERT**, **INTO**, **VALUES**, **UPDATE**, **SET**, **AND**.

Nous allons étudier une base de données traitant du cinéma dont voici le schéma relationnel qui comporte 3 relations :

- la relation **individu** (id_ind, nom, prenom, naissance)
- la relation **realisation** (id_rea, titre, annee, type)
- la relation **emploi** (id_emp, description, #id_ind, #id_rea)

Les clés primaires sont soulignées et les clés étrangères sont précédées d'un #.

Ainsi **emploi.id_ind** est une clé étrangère faisant référence à **individu.id_ind**.

Voici un extrait des tables **individu** et **realisation** :

extrait de individu				extrait de realisation			
id_ind	nom	prenom	naissance	id_rea	titre	annee	type
105	'Hulka'	'Daniel'	'01-06-1968'	105	'Casino Imperial'	2006	'action'
403	'Travis'	'Daniel'	'10-03-1968'	325	'Ciel tombant'	2012	'action'
688	'Crog'	'Daniel'	'07-07-1968'	655	'Fantôme'	2015	'action'
695	'Pollock'	'Daniel'	'24-08-1968'	950	'Mourir pour attendre'	2021	'action'

1. On s'intéresse ici à la récupération de données dans une relation.

a. Écrire ce que renvoie la requête ci-dessous :

```
SELECT nom, prenom, naissance
FROM individu
WHERE nom = 'Crog';
```

b. Fournir une requête SQL permettant de récupérer le titre et la clé primaire de chaque film dont la date de sortie est strictement supérieure à 2020.

2. Cette question traite de la modification de relations.

a. Dire s'il faut utiliser la requête 1 ou la requête 2 proposées ci-dessous pour modifier la date de naissance de Daniel Crog. Justifier votre réponse en expliquant pourquoi la requête refusée ne pourra pas fonctionner.

```
UPDATE individu Requête 1
SET naissance = '02-03-1968'
WHERE id_ind = 688 AND nom = 'Crog' AND prenom = 'Daniel';
```

```
INSERT INTO individu Requête 2
VALUES (688, 'Crog', 'Daniel', '02-03-1968');
```

- b. Expliquer si la relation **individu** peut accepter (ou pas) deux individus portant le même nom, le même prénom et la même date de naissance.
3. Cette question porte sur la notion de clés étrangères.
- a. Recopier sur votre copie les demandes ci-dessous, dans leur intégralité, et les compléter correctement pour qu'elles ajoutent dans la relation **emploi** les rôles de Daniel Crog en tant que James Bond dans le film nommé 'Casino Impérial' puis dans le film 'Ciel tombant'.

```
INSERT INTO emploi
VALUES (5400, 'Acteur(James Bond)', ... );

INSERT INTO emploi
VALUES (5401, 'Acteur(James Bond)', ... );
```

- b. On désire rajouter un nouvel emploi de Daniel Crog en tant que James Bond dans le film 'Docteur Yes'.
Expliquer si l'on doit d'abord créer l'enregistrement du film dans la relation **realisation** ou si l'on doit d'abord créer le rôle dans la relation **emploi**.
4. Cette question traite des jointures.
- a. Recopier sur votre copie la requête SQL ci-dessous, dans son intégralité, et la compléter de façon à ce qu'elle renvoie le nom de l'acteur, le titre du film et l'année de sortie du film, à partir de tous les enregistrements de la relation **emploi** pour lesquels la description de l'emploi est 'Acteur(James Bond) '.

```
SELECT ...
FROM emploi
JOIN individu ON ...
JOIN realisation ON ...
WHERE emploi.description = 'Acteur(James Bond) ';
```

- b. Fournir une requête SQL permettant de trouver toutes les descriptions des emplois de Denis Johnson (Denis est son prénom et Johnson est son nom). On veillera à n'afficher que la description des emplois et non les films associés à ces emplois.

EXERCICE 3 (4 points)

Cet exercice porte sur les réseaux et les protocoles de routages.

Rappels :

Une adresse IPv4 est composée de 4 octets, soit 32 bits. Elle est notée a.b.c.d, où a, b, c et d sont les valeurs des 4 octets.

La notation a.b.c.d/n signifie que les n premiers bits de l'adresse IP représentent la partie « réseau », les bits qui suivent représentent la partie « machine ».

L'adresse IPv4 dont tous les bits de la partie « machine » sont à 0 est appelée « adresse du réseau ».

L'adresse IPv4 dont tous les bits de la partie « machine » sont à 1 est appelée « adresse de diffusion ».

On considère le réseau représenté sur la Figure 1 ci-dessous :

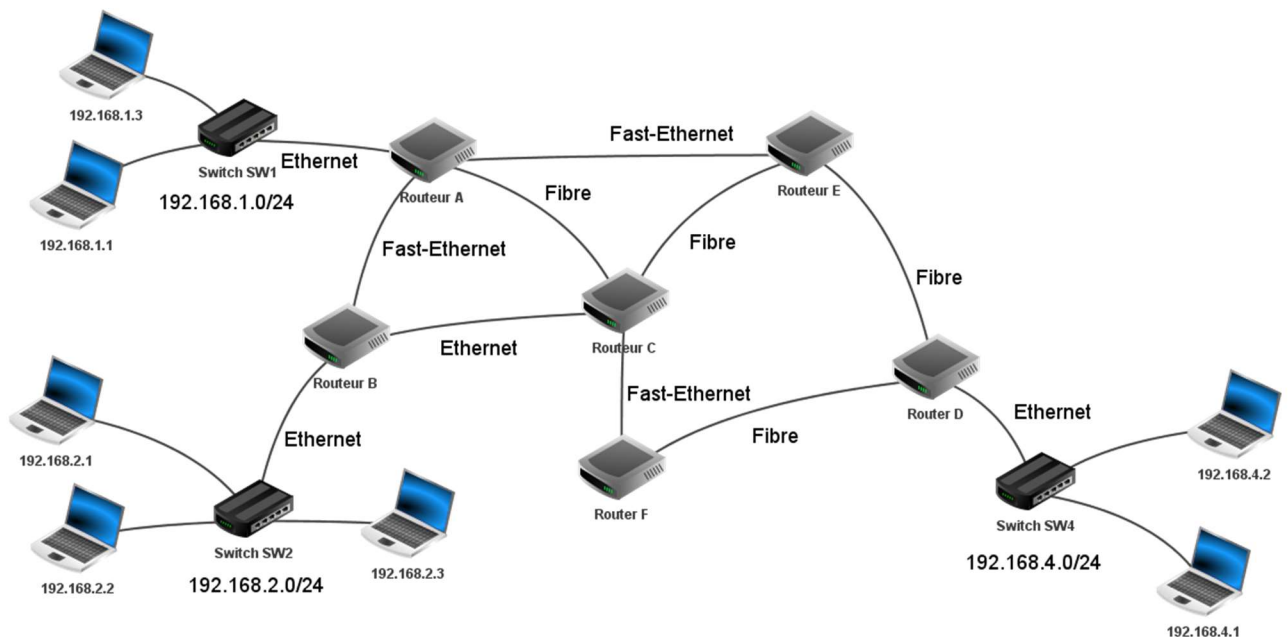


Figure 1 : schéma du réseau

1. On considère la machine d'adresse IPv4 192.168.1.1/24
 - a. Donner l'adresse du réseau sur lequel se trouve cette machine.
 - b. Donner l'adresse de diffusion (*broadcast*) de ce réseau.
 - c. Donner le nombre maximal de machines que l'on peut connecter sur ce réseau.
 - d. On souhaite ajouter une machine sur ce réseau, proposer une adresse IPv4 possible pour cette machine.
2.
 - a. La machine d'adresse IPv4 192.168.1.1 transmet un paquet IPv4 à la machine d'adresse IPv4 192.168.4.2
Donner toutes les routes pouvant être empruntées par ce paquet IPv4, chaque routeur ne pouvant être traversé qu'une seule fois.

- b. Expliquer l'utilité d'avoir plusieurs routes possibles reliant les réseaux 192.168.1.0/24 et 192.168.4.0/24
3. Dans cette question, on suppose que le protocole de routage mis en place dans le réseau est RIP. Ce protocole consiste à minimiser le nombre de sauts. Le schéma du réseau est celui de la figure 1.
Les tables de routage utilisées sont données ci-dessous :

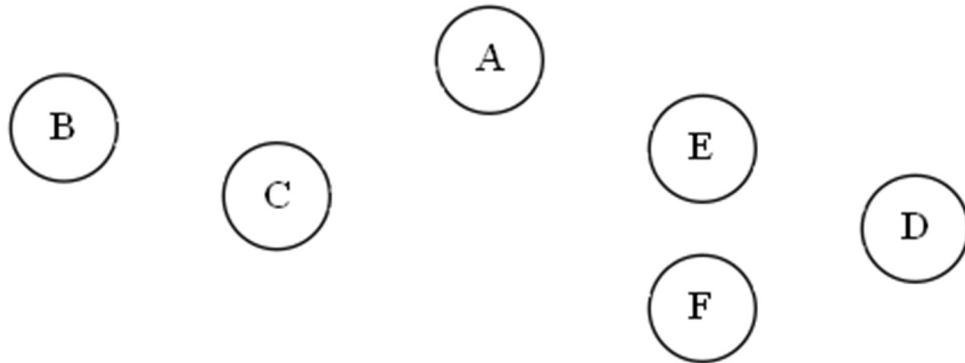
Routeur A		Routeur B		Routeur C	
Destination	passé par	Destination	passé par	Destination	passé par
B	...	A	A	A	A
C	...	C	C	B	B
D	E	D	C	D	E
E	...	E	C	E	E
F	C	F	C	F	F

Routeur D		Routeur E		Routeur F	
Destination	passé par	Destination	passé par	Destination	passé par
A	E	A	A	A	C
B	F	B	C	B	C
C	F	C	C	C	C
E	E	D	D	D	D
F	F	F	C	E	C

Tables de routage

- a. Recopier et compléter sur la copie la table de routage du routeur A.
- b. Un paquet IP doit aller du routeur B au routeur D. En utilisant les tables de routage, donner le parcours emprunté par celui-ci.
- c. Les connexions entre les routeurs B-C et A-E étant coupées, sur la copie, réécrire les tables de routage des routeurs A, B et C.
- d. Déterminer le nouveau parcours emprunté par le paquet IP pour aller du routeur B au routeur D.
4. Dans cette question, on suppose que le protocole de routage mis en place dans le réseau est OSPF. Ce protocole consiste à minimiser la somme des coûts des liaisons empruntées. Le coût d'une liaison est défini par la relation $\text{coût} = \frac{10^8}{d}$ où d représente le débit en bit/s et coût est sans unité. Le schéma du réseau est celui de la figure 1.

- a. Déterminer le coût des liaisons Ethernet ($d = 10^7$ bit/s), Fast-Ethernet ($d = 10^8$ bit/s) et Fibre ($d = 10^9$ bit/s).
- b. On veut représenter schématiquement le réseau de routeurs à partir du schéma du réseau figure 1.
Recopier sur la copie le schéma ci-dessous et tracer les liaisons entre les routeurs en y indiquant le coût.



- c. Un paquet IPv4 doit être acheminé d'une machine ayant pour adresse IPv4 192.168.2.1 à une machine ayant pour adresse IPv4 192.168.4.1
Écrire les routes possibles, c'est à dire la liste des routeurs traversés, et le coût de chacune de ces routes, chaque routeur ne pouvant être traversé qu'une seule fois.
- d. Donner, en la justifiant, la route qui sera empruntée par un paquet IPv4 pour aller d'une machine ayant pour adresse IPv4 192.168.2.1 à une machine ayant pour adresse IPv4 192.168.4.1