

Exercice 1 (8 points)

Cet exercice porte sur les arbres binaires, la programmation orienté objet et la récursivité

Dans un arbre binaire, chaque nœud admet au plus deux enfants, appelés sous-arbre gauche et sous-arbre droit. On considère dans cet exercice des arbres binaires étiquetés avec des nombres entiers.

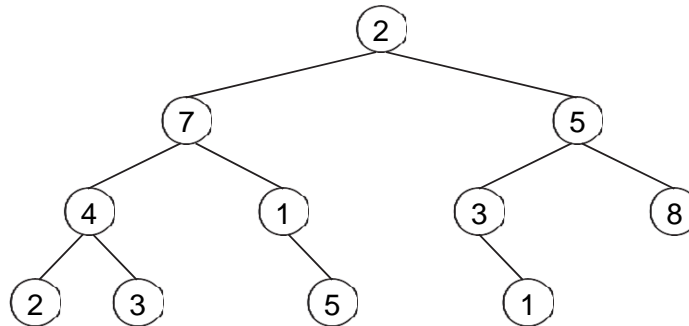
On définit un chemin racine-feuille dans un tel arbre comme une liste ordonnée de nœuds telle que

- le premier nœud est la racine ;
- chaque nœud suivant est enfant du précédent ;
- le dernier nœud est une feuille.

On appellera somme d'un chemin racine-feuille la somme des étiquettes des nœuds du chemin.

Enfin, la plus grande somme racine-feuille d'un arbre est la plus grande somme qu'il est possible d'obtenir en considérant tous les chemins racine-feuille de l'arbre.

1. Déterminer la plus grande somme racine-feuille de l'arbre représenté ci-dessous.



2. La classe `Noeud` ci-dessous implémente le type abstrait d'arbre binaire.

```

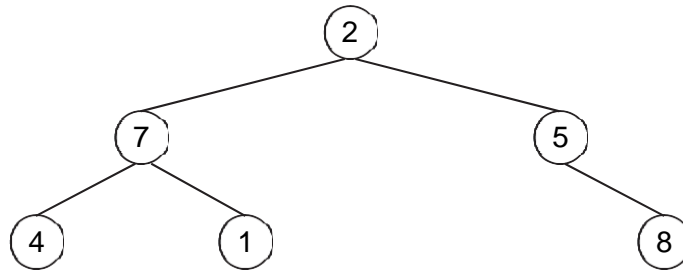
class Noeud:
    def __init__(self, v):
        self.etiquette = v
        self.sag = None
        self.sad = None

    def niveau(self):
        if self.sag!=None and self.sad!=None:
            hg = self.sag.niveau()
            hd = self.sad.niveau()
            return 1+max(hg, hd)
        if self.sag!=None:
            return self.sag.niveau()+1
        if self.sad!=None:
            return self.sad.niveau()+1
        return 0

    def modifier_sag(self, nsag) :
        self.sag = nsag

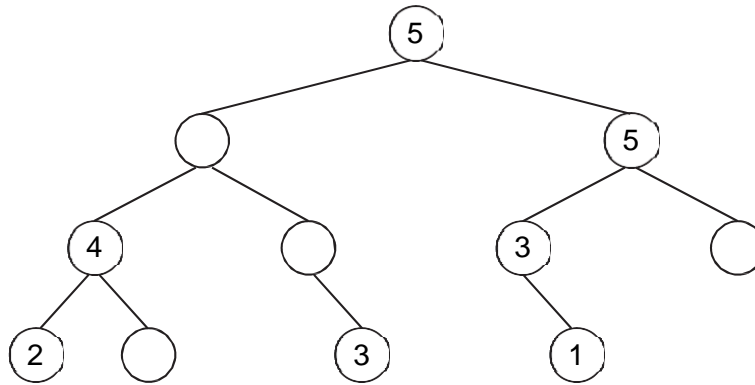
    def modifier_sad(self, nsad) :
        self.sad = nsad
  
```

- a. Écrire une suite d'instructions utilisant la classe `Noeud` permettant de représenter l'arbre ci-dessous.



- b. Que renvoie l'appel de la méthode `niveau` sur l'arbre ci-dessus ?
3. S'inspirer du code de la méthode `niveau` pour écrire une méthode récursive `pgde_somme` qui renvoie la plus grande somme racine-feuille d'un arbre.
4. On appelle arbre magique un arbre binaire dont toutes les sommes des chemins racine-feuille sont égales.

- a. Recopier et compléter l'arbre ci-dessous pour qu'il soit magique.

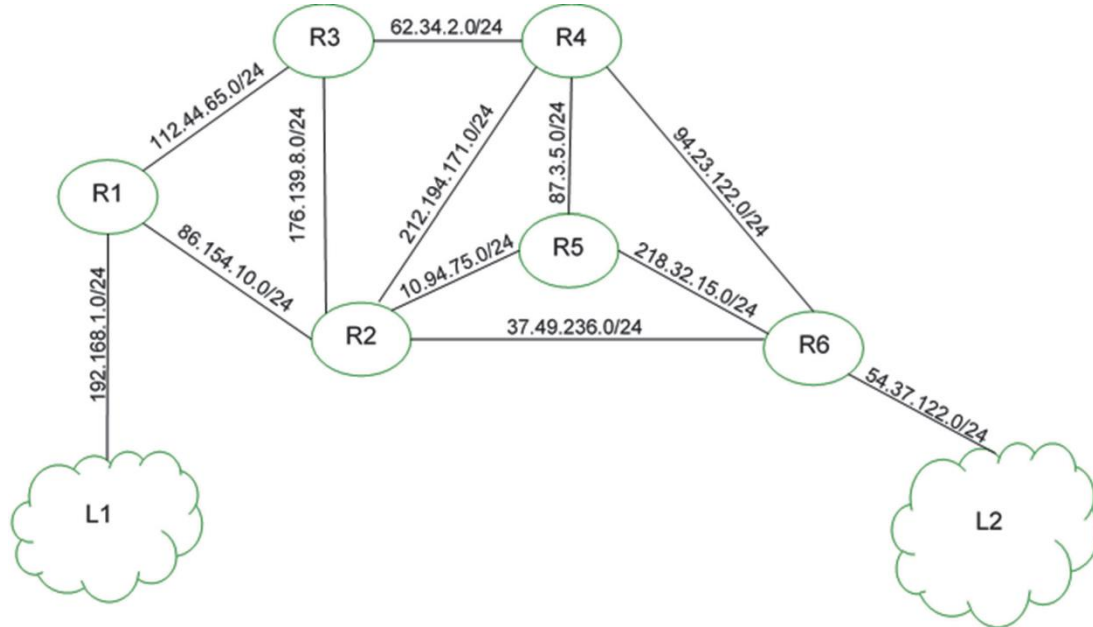


- b. Un arbre est magique si ses sous-arbres sont magiques et qu'ils ont de plus la même plus grande somme racine-feuille. Écrire une méthode récursive `est_magique` qui renvoie `True` si l'arbre est magique et `False` sinon.

EXERCICE 2 (8 points)

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

On représente ci-dessous un réseau dans lequel R1, R2, R3, R4, R5 et R6 sont des routeurs. Le réseau local L1 est relié au routeur R1 et le réseau local L2 au routeur R6.



Rappels et notations

Dans cet exercice, les adresses IP sont composées de 4 octets, soit 32 bits. Elles sont notées X1.X2.X3.X4, où X1, X2, X3 et X4 sont les valeurs des 4 octets, convertis en notation décimale.

La notation X1.X2.X3.X4/n signifie que les n premiers bits de poids forts de l'adresse IP représentent la partie « réseau », les bits suivants représentent la partie « hôte ».

Toutes les adresses des hôtes connectés à un réseau local ont la même partie réseau et peuvent donc communiquer directement. L'adresse IP dont tous les bits de la partie « hôte » sont à 0 est appelée « adresse du réseau ».

On donne également des extraits de la table de routage des routeurs R1 à R5 dans le tableau suivant :

Routeur	Réseau destinataire	Passerelle	Interface
R1	54.37.122.0/24	86.154.10.1	86.154.10.56
R2	54.37.122.0/24	37.49.236.22	37.49.236.23
R3	54.37.122.0/24	62.34.2.8	62.34.2.9
R4	54.37.122.0/24	94.23.122.10	94.23.122.11

R5	54.37.122.0/24	218.32.15.1	218.32.15.2
----	----------------	-------------	-------------

1. Un paquet part du réseau local L1 à destination du réseau local L2.
 - a. En utilisant l'extrait de la table de routage de R1, vers quel routeur R1 envoie-t-il ce paquet : R2 ou R3 ? Justifier.

Le routeur R1 envoie le paquet vers le routeur R2 car dans la table de routage l'adresse IP de la passerelle indiquée est 86.154.10.1 et la liaison entre R1 et R2 l'adresse IP indiquée est 86.154.10.0/24 /24 signifie que 86.154.10 est la partie réseau commune aux deux adresses IP 86.154.10.1 et 86.154.10.0.

- b. A l'aide des extraits de tables de routage ci-dessus, nommer les routeurs traversés par ce paquet, lorsqu'il va du réseau L1 au réseau L2.

Pour aller du réseau L1 au réseau L2, on passe par les routeurs R1, R2 et R6.

2. La liaison entre R1 et R2 est rompue.
 - a. Sachant que ce réseau utilise le protocole RIP (distance en nombre de sauts), donner l'un des deux chemins possibles que pourra suivre un paquet allant de L1 vers L2.

Les 2 chemins possibles avec RIP sont :

- R1 – R3 – R4 – R6
- R1 – R3 – R2 – R6

avec 3 sauts

- b. Dans les extraits de tables de routage ci-dessus, pour le chemin de la question 2.a, quelle(s) ligne(s) sera (seront) modifiée(s) ?

La ligne modifiée sera celle du routeur R1 :

R1	54.37.122.0/24	112.144.65.1	86.154.10.56
----	----------------	--------------	--------------

3. On a rétabli la liaison entre R1 et R2. Par ailleurs, pour tenir compte du débit des liaisons, on décide d'utiliser le protocole OSPF (distance liée au coût minimal des liaisons) pour effectuer le routage. Le coût des liaisons entre les routeurs est donné par le tableau suivant :

Liaison	R1-R2	R1-R3	R2-R3	R2-R4	R2-R5	R2-R6	R3-R4	R4-R5	R4-R6	R5-R6
Coût	100	100	?	1	10	10	10	1	10	1

a. Le coût C d'une liaison est donné ici par la formule

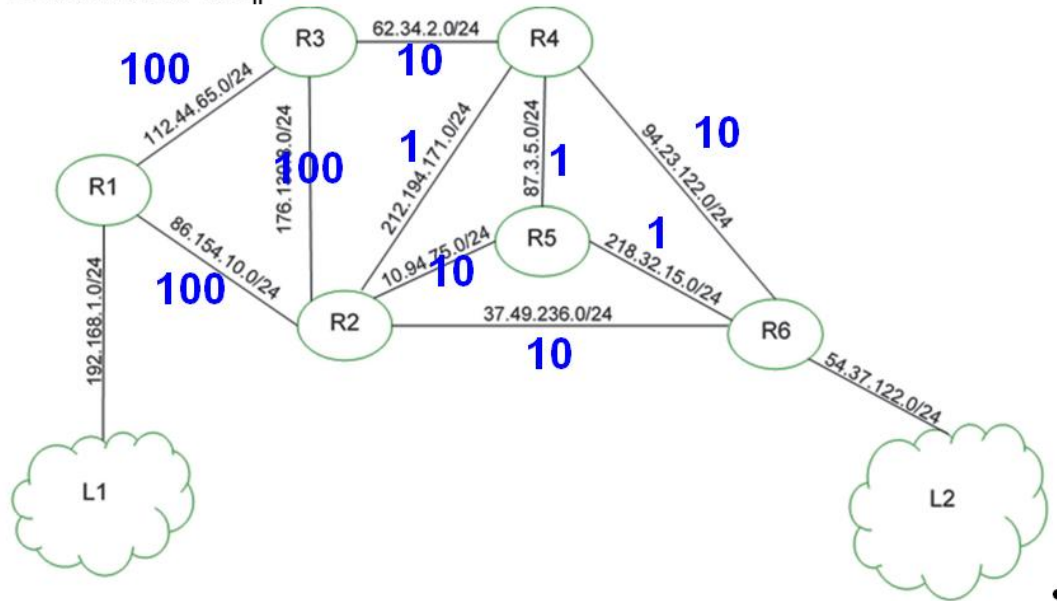
$$C = \frac{10^9}{BP}$$

où BP est la bande passante de la connexion en bps (bit par seconde). Sachant que la bande passante de la liaison R2-R3 est de 10 Mbps, calculer le coût correspondant.

Le coût est $C = 10^9 / (10 \times 10^6) = 100$

b. Déterminer le chemin parcouru par un paquet partant du réseau L1 et arrivant au réseau L2, en utilisant le protocole OSPF.

c. Le schéma ci-dessous :



Le chemin avec OSPF correspond au coût le plus faible :

Soit R1 – R2 – R4 – R5 – R6 avec un coût de : $100 + 1 + 1 + 1 = 103$

d. Indiquer pour quel(s) routeur(s) l'extrait de la table de routage sera modifié pour un paquet à destination de L2, avec la métrique OSPF.

Les routeurs suivants seront modifiés dans la table de routage avec la métrique OSPF :

R2	54.37.122.0/24	212.194.171.0	37.49.236.23
R4	54.37.122.0/24	87.3.5.0	94.23.122.11

EXERCICE 3 (4 points)

Cet exercice porte sur la gestion des processus par les systèmes d'exploitation et sur les opérateurs booléens.

Pour une meilleure lisibilité, des espaces sont placées dans les écritures binaires des nombres. Il ne faut pas les prendre en compte dans les calculs.

Pour chiffrer un message, une méthode, dite du masque jetable, consiste à le combiner avec une chaîne de caractères de longueur comparable.

Une implémentation possible utilise l'opérateur XOR (ou exclusif) dont voici la table de vérité :

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Dans la suite, les nombres écrits en binaire seront précédés du préfixe 0b.

1. Pour chiffrer un message, on convertit chacun de ses caractères en binaire (à l'aide du format Unicode), et on réalise l'opération XOR bit à bit avec la clé.

Après conversion en binaire, et avant que l'opération XOR bit à bit avec la clé n'ait été effectuée, Alice obtient le message suivant :

$$m = 0b\ 0110\ 0011\ 0100\ 0110$$

- a. Le message m correspond à deux caractères codés chacun sur 8 bits : déterminer quels sont ces caractères. On fournit pour cela la table ci-dessous qui associe à l'écriture hexadécimale d'un octet le caractère correspondant (figure 2). Exemple de lecture : le caractère correspondant à l'octet codé 4A en hexadécimal est la lettre J.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	space	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Figure 2

- b. Pour chiffrer le message d'Alice, on réalise l'opération XOR bit à bit avec la clé suivante :

$$k = 0b\ 1110\ 1110\ 1111\ 0000$$

Donner l'écriture binaire du message obtenu.

2.

- a. Dresser la table de vérité de l'expression booléenne suivante :

$$(a \text{ XOR } b) \text{ XOR } b$$

- b. Bob connaît la chaîne de caractères utilisée par Alice pour chiffrer le message. Quelle opération doit-il réaliser pour déchiffrer son message ?

Exercice 1 (8 points)

Cet exercice porte sur les arbres binaires, la programmation orienté objet et la récursivité

Dans un arbre binaire, chaque nœud admet au plus deux enfants, appelés sous-arbre gauche et sous-arbre droit. On considère dans cet exercice des arbres binaires étiquetés avec des nombres entiers.

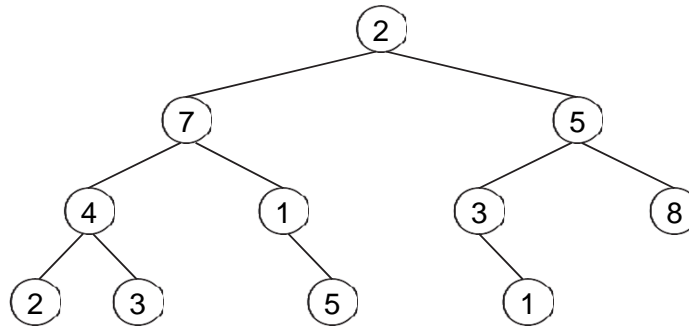
On définit un chemin racine-feuille dans un tel arbre comme une liste ordonnée de nœuds telle que

- le premier nœud est la racine ;
- chaque nœud suivant est enfant du précédent ;
- le dernier nœud est une feuille.

On appellera somme d'un chemin racine-feuille la somme des étiquettes des nœuds du chemin.

Enfin, la plus grande somme racine-feuille d'un arbre est la plus grande somme qu'il est possible d'obtenir en considérant tous les chemins racine-feuille de l'arbre.

1. Déterminer la plus grande somme racine-feuille de l'arbre représenté ci-dessous.



La somme de chaque branche est :

- $2+7+4+2 = 15$
- $2+7+4+3 = 16$
- $2+7+1+5 = 14$
- $2+5+3+1 = 11$
- $2+5+8 = 15$

La plus grande somme racine-feuille est donc 16.

2. La classe `Noeud` ci-dessous implémente le type abstrait d'arbre binaire.

```
class Noeud:

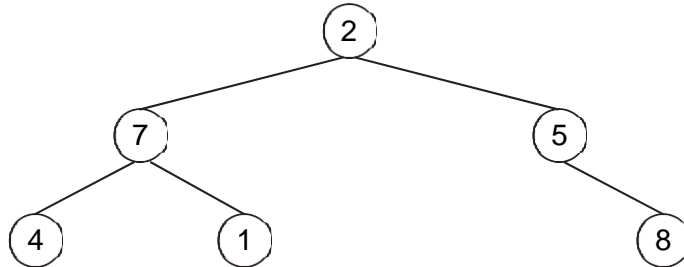
    def __init__(self, v):
        self.etiquette = v
        self.sag = None
        self.sad = None

    def niveau(self):
        if self.sag!=None and self.sad!=None:
            hg = self.sag.niveau()
            hd = self.sad.niveau()
            return 1+max(hg, hd)
        if self.sag!=None:
            return self.sag.niveau()+1
        if self.sad!=None:
            return self.sad.niveau()+1
        return 0

    def modifier_sag(self, nsag) :
        self.sag = nsag

    def modifier_sad(self, nsad) :
        self.sad = nsad
```

a. Écrire une suite d'instructions utilisant la classe `Noeud` permettant de représenter l'arbre ci-dessous.



```
n4 = Noeud(4)
n1 = Noeud(1)
n7 = Noeud(7)
n7.modifier_sad(n4)
n7.modifier_sad(n1)
n8 = Noeud(8)
n5 = Noeud(5)
n5.modifier_sad(n8)
n2 = Noeud(2)
n2.modifier_sad(n5)
n2.modifier_sag(n7)
```

b. Que renvoie l'appel de la méthode `niveau` sur l'arbre ci-dessus ?

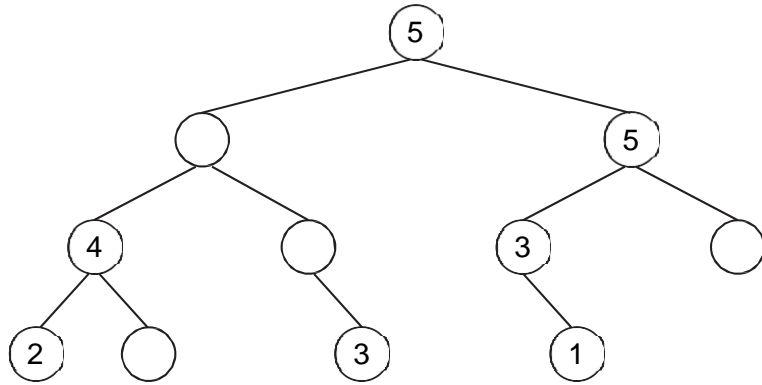
L'appel de la méthode `niveau` sur l'arbre ci-dessus renvoie 2.
La méthode `niveau()` renvoie la hauteur de l'arbre.

3. S'inspirer du code de la méthode `niveau` pour écrire une méthode récursive `pgde_somme` qui renvoie la plus grande somme racine-feuille d'un arbre.

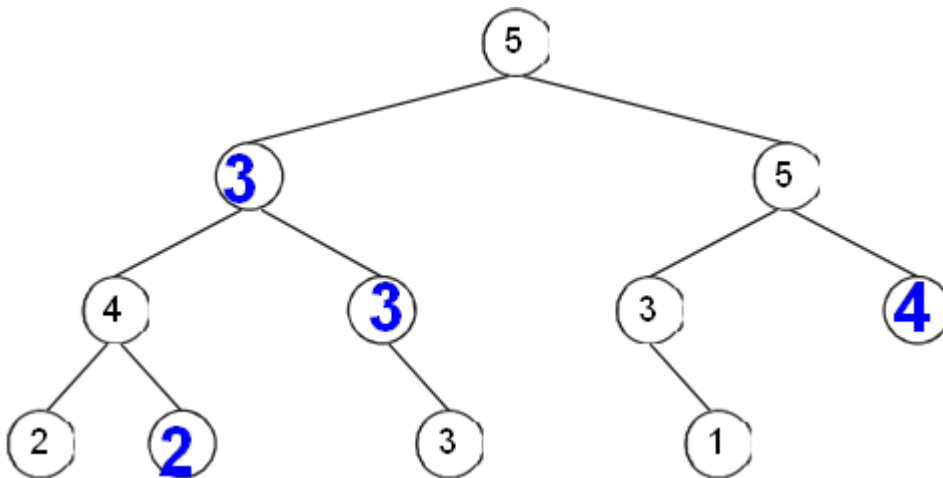
```
def pgde_somme(self):
    if self.sag is not None and self.sad is not None:
        somme_gauche = self.sag.pgde_somme()
        somme_droite = self.sad.pgde_somme()
        return self.etiquette + max(somme_gauche,somme_droite)
    if self.sag is not None:
        return self.sag.pgde_somme() + self.etiquette
    if self.sad is not None:
        return self.sad.pgde_somme() + self.etiquette
    return self.etiquette
```

4. On appelle arbre magique un arbre binaire dont toutes les sommes des chemins racine-feuille sont égales.

- a. Recopier et compléter l'arbre ci-dessous pour qu'il soit magique.



La somme magique est $5 + 5 + 3 + 1 + 14$.



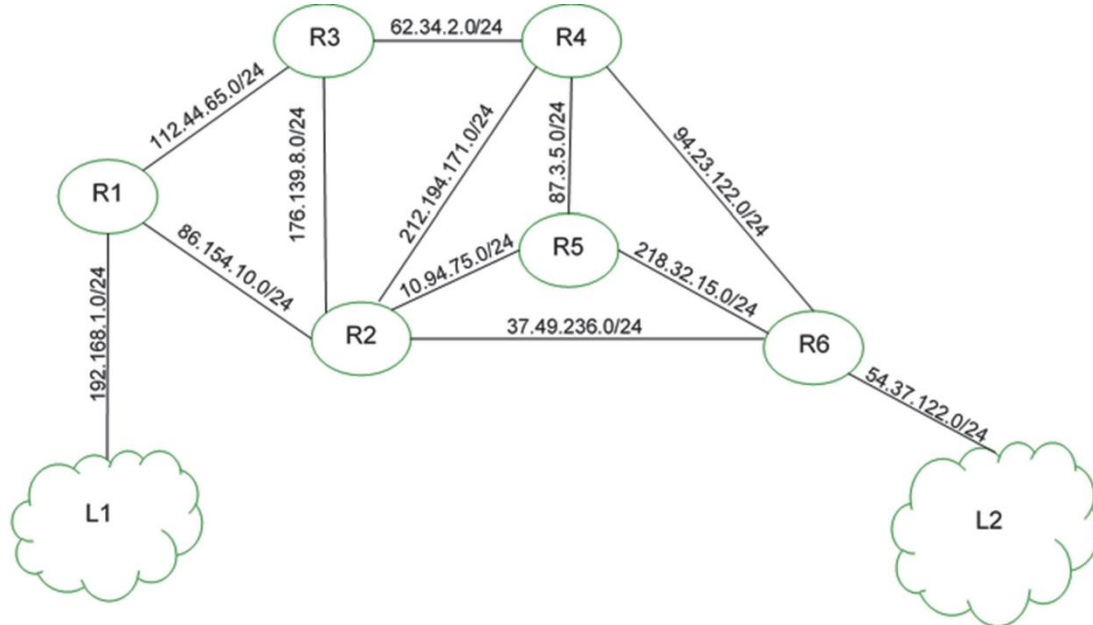
- b. Un arbre est magique si ses sous-arbres sont magiques et qu'ils ont de plus la même plus grande somme racine-feuille. Écrire une méthode récursive `est_magique` qui renvoie `True` si l'arbre est magique et `False` sinon.

```
def est_magique(self):
    if self.sag is not None and self.sad is not None:
        somme_max_sag = self.sag.pgde_somme()
        somme_max_sad = self.sad.pgde_somme()
        return self.sag.est_magique() and self.sad.est_magique()
    and somme_max_sag == somme_max_sad
    if self.sag is not None:
        return self.sag.est_magique()
    if self.sad is not None:
        return self.sad.est_magique()
    return True
```

EXERCICE 2 (8 points)

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

On représente ci-dessous un réseau dans lequel R1, R2, R3, R4, R5 et R6 sont des routeurs. Le réseau local L1 est relié au routeur R1 et le réseau local L2 au routeur R6.



Rappels et notations

Dans cet exercice, les adresses IP sont composées de 4 octets, soit 32 bits. Elles sont notées X1.X2.X3.X4, où X1, X2, X3 et X4 sont les valeurs des 4 octets, convertis en notation décimale.

La notation X1.X2.X3.X4/n signifie que les n premiers bits de poids forts de l'adresse IP représentent la partie « réseau », les bits suivants représentent la partie « hôte ».

Toutes les adresses des hôtes connectés à un réseau local ont la même partie réseau et peuvent donc communiquer directement. L'adresse IP dont tous les bits de la partie « hôte » sont à 0 est appelée « adresse du réseau ».

On donne également des extraits de la table de routage des routeurs R1 à R5 dans le tableau suivant :

Routeur	Réseau destinataire	Passerelle	Interface
R1	54.37.122.0/24	86.154.10.1	86.154.10.56
R2	54.37.122.0/24	37.49.236.22	37.49.236.23
R3	54.37.122.0/24	62.34.2.8	62.34.2.9

CORRECTION

R4	54.37.122.0/24	94.23.122.10	94.23.122.11
R5	54.37.122.0/24	218.32.15.1	218.32.15.2

1. Un paquet part du réseau local L1 à destination du réseau local L2.
 - a. En utilisant l'extrait de la table de routage de R1, vers quel routeur R1 envoie-t-il ce paquet : R2 ou R3 ? Justifier.
 - b. A l'aide des extraits de tables de routage ci-dessus, nommer les routeurs traversés par ce paquet, lorsqu'il va du réseau L1 au réseau L2.
2. La liaison entre R1 et R2 est rompue.
 - a. Sachant que ce réseau utilise le protocole RIP (distance en nombre de sauts), donner l'un des deux chemins possibles que pourra suivre un paquet allant de L1 vers L2.
 - b. Dans les extraits de tables de routage ci-dessus, pour le chemin de la question 2.a, quelle(s) ligne(s) sera (seront) modifiée(s) ?
3. On a rétabli la liaison entre R1 et R2.
 Par ailleurs, pour tenir compte du débit des liaisons, on décide d'utiliser le protocole OSPF (distance liée au coût minimal des liaisons) pour effectuer le routage. Le coût des liaisons entre les routeurs est donné par le tableau suivant :

Liaison	R1-R2	R1-R3	R2-R3	R2-R4	R2-R5	R2-R6	R3-R4	R4-R5	R4-R6	R5-R6
Coût	100	100	?	1	10	10	10	1	10	1

- a. Le coût C d'une liaison est donné ici par la formule

$$C = \frac{10^9}{BP}$$

où BP est la bande passante de la connexion en bps (bit par seconde). Sachant que la bande passante de la liaison R2-R3 est de 10 Mbps, calculer le coût correspondant.

- b. Déterminer le chemin parcouru par un paquet partant du réseau L1 et arrivant au réseau L2, en utilisant le protocole OSPF.
- c. Indiquer pour quel(s) routeur(s) l'extrait de la table de routage sera modifié pour un paquet à destination de L2, avec la métrique OSPF.

EXERCICE 3 (4 points)

Cet exercice porte sur la gestion des processus par les systèmes d'exploitation et sur les opérateurs booléens.

Pour une meilleure lisibilité, des espaces sont placées dans les écritures binaires des nombres. Il ne faut pas les prendre en compte dans les calculs.

Pour chiffrer un message, une méthode, dite du masque jetable, consiste à le combiner avec une chaîne de caractères de longueur comparable.

Une implémentation possible utilise l'opérateur XOR (ou exclusif) dont voici la table de vérité :

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Dans la suite, les nombres écrits en binaire seront précédés du préfixe 0b.

1. Pour chiffrer un message, on convertit chacun de ses caractères en binaire (à l'aide du format Unicode), et on réalise l'opération XOR bit à bit avec la clé.

Après conversion en binaire, et avant que l'opération XOR bit à bit avec la clé n'ait été effectuée, Alice obtient le message suivant :

$$m = 0b\ 0110\ 0011\ 0100\ 0110$$

- a. Le message m correspond à deux caractères codés chacun sur 8 bits : déterminer quels sont ces caractères. On fournit pour cela la table ci-dessous qui associe à l'écriture hexadécimale d'un octet le caractère correspondant (figure 2). Exemple de lecture : le caractère correspondant à l'octet codé 4A en hexadécimal est la lettre J.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	space	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Figure 2

0110 0011 = 1+2+32+64 = 99 en décimal
 = 16 x 6 + 3 = 63 en hexadécimal
 Ce qui correspond à la lettre 'c' dans la table.

0100 0110 = 2 + 4 + 64 = 70 en décimal
 = 16 x 4 + 6 = 46 en hexadécimal

Ce qui correspond à la lettre 'F' dans la table.
 On a donc m = 'cF'

- b. Pour chiffrer le message d'Alice, on réalise l'opération XOR bit à bit avec la clé suivante :

k = 0b 1110 1110 1111 0000

Donner l'écriture binaire du message obtenu.

```
0110 0011 0100 0110
1110 1110 1111 0000
-----
1000 1101 1011 0110
```

2.

- a. Dresser la table de vérité de l'expression booléenne suivante :
 (a XOR b) XOR b

a	b	a XOR b	(a XOR b) XOR b
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

- b. Bob connaît la chaîne de caractères utilisée par Alice pour chiffrer le message. Quelle opération doit-il réaliser pour déchiffrer son message ?

On remarque d'après la question précédente que : $(a \text{ XOR } b) \text{ XOR } b = a$

Donc Bob doit effectuer un XOR du message chiffré avec la clef k utilisée par Alice pour déchiffrer le message.