

# Evaluation théorique N°4

Durée : 1 heure

La calculatrice est interdite ainsi que l'utilisation de tout document.

Rappel : Parmi les deux exercices suivants, en choisir UN seul !

## Exercice 1 :

*Cet exercice traite du thème « algorithmique », et principalement des algorithmes sur les arbres binaires.*

On manipule ici les arbres binaires avec trois fonctions :

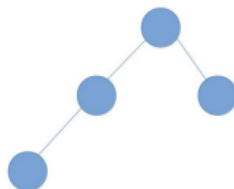
- `est_vide(A)` qui renvoie `True` si l'arbre binaire `A` est vide, `False` s'il ne l'est pas ;
- `sous_arbre_gauche(A)` qui renvoie le sous-arbre gauche de l'arbre binaire `A` ;
- `sous_arbre_droit(A)` qui renvoie le sous-arbre droit de l'arbre binaire `A`.

L'arbre binaire renvoyé par les fonctions `sous_arbre_gauche` et `sous_arbre_droit` peut éventuellement être l'arbre vide.

On définit la **hauteur** d'un arbre binaire non vide de la façon suivante :

- si ses sous-arbres gauche et droit sont vides, sa hauteur est 0 ;
- si l'un des deux au moins est non vide, alors sa hauteur est égale à  $1 + M$ , où  $M$  est la plus grande des hauteurs de ses sous-arbres (gauche et droit) non vides.

1. a. Donner la hauteur de l'arbre ci-dessous.



b. Dessiner sur la copie un arbre binaire de hauteur 4.

La hauteur d'un arbre est calculée par l'algorithme récursif suivant :

```
1  Algorithme hauteur(A) :
2    test d'assertion : A est supposé non vide
3    si sous_arbre_gauche(A) vide et sous_arbre_droit(A) vide:
4      renvoyer 0
5    sinon, si sous_arbre_gauche(A) vide:
6      renvoyer 1 + hauteur(sous_arbre_droit(A))
7    sinon, si ... :
8      renvoyer ...
9    sinon:
10     renvoyer 1 + max(hauteur(sous_arbre_gauche(A)),
11                      hauteur(sous_arbre_droit(A)))
```

2. Recopier sur la copie les lignes 7 et 8 en complétant les points de suspension.
3. On considère un arbre binaire  $R$  dont on note  $G$  le sous-arbre gauche et  $D$  le sous-arbre droit. On suppose que  $R$  est de hauteur 4 et  $G$  de hauteur 2.
  - a. Justifier le fait que  $D$  n'est pas l'arbre vide et déterminer sa hauteur.
  - b. Illustrer cette situation par un dessin.

Soit un arbre binaire non vide de hauteur  $h$ . On note  $n$  le nombre de nœuds de cet arbre. On admet que  $h+1 \leq n \leq 2^{h+1}-1$ .

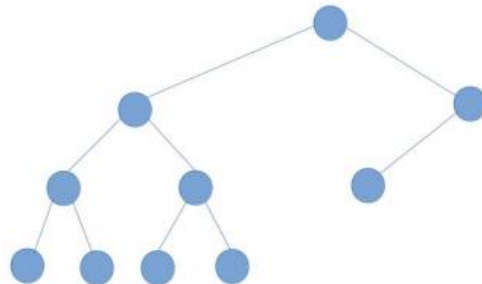
4. a. Vérifier ces inégalités sur l'arbre binaire de la question 1.a.
  - b. Expliquer comment construire un arbre binaire de hauteur  $h$  quelconque ayant  $h+1$  nœuds.
  - c. Expliquer comment construire un arbre binaire de hauteur  $h$  quelconque ayant  $2^{h+1}-1$  nœuds.
- Indication :*  $2^{h+1}-1 = 1+2+4+\dots+2^h$ .

L'objectif de la fin de l'exercice est d'écrire le code d'une fonction `fabrique(h, n)` qui prend comme paramètres deux nombres entiers positifs  $h$  et  $n$  tels que  $h+1 < n < 2^{h+1}-1$ , et qui renvoie un arbre binaire de hauteur  $h$  à  $n$  nœuds.

Pour cela, on a besoin des deux fonctions suivantes:

- `arbre_vide()`, qui renvoie un arbre vide ;
- `arbre(gauche, droit)` qui renvoie l'arbre de fils gauche `gauche` et de fils droit `droit`.

5. Recopier sur la copie l'arbre binaire ci-dessous et numéroter ses nœuds de 1 en 1 en commençant à 1, en effectuant un parcours en profondeur préfixe.



La fonction `fabrique` ci-dessous a pour but de répondre au problème posé. Pour cela, la fonction `annexe` utilise la valeur de `n`, qu'elle peut modifier, et renvoie un arbre binaire de hauteur `hauteur_max` dont le nombre de nœuds est égal à la valeur de `n` au moment de son appel.

```
1. def fabrique(h, n):
2.     def annexe(hauteur_max):
3.         if n == 0 :
4.             return arbre_vide()
5.         elif hauteur_max == 0:
6.             n = n - 1
7.             return ...
8.         else:
9.             n = n - 1
10.            gauche = annexe(hauteur_max - 1)
11.            droite = ...
12.            return arbre(gauche, droite)
13.    return annexe(h)
```

6. Recopier sur la copie les lignes 7 et 11 en complétant les points de suspension.

## Exercice 2 :

Cet exercice porte sur les représentations binaires et les protocoles de routage.

1. Une adresse IPv4 est représentée sous la forme de 4 nombres séparés par des points. Chacun de ces 4 nombres peut être représenté sur un octet.
  - a. Donner en écriture décimale l'adresse IPv4 correspondant à l'écriture binaire : 11000000.10101000.10000000.10000011
  - b. Tous les ordinateurs du réseau A ont une adresse IPv4 de la forme : 192.168.128.\_\_\_ , où seul le dernier octet (représenté par \_\_\_ ) diffère. Donner le nombre d'adresses différentes possibles du réseau A.
2. On rappelle que le protocole RIP cherche à minimiser le nombre de routeurs traversés (qui correspond à la métrique). On donne les tables de routage d'un réseau informatique composé de 5 routeurs (appelés A, B, C, D et E), chacun associé directement à un réseau du même nom obtenues avec le protocole RIP :

Routeur A

Destination	Métrique
A	0
B	1
C	1
D	1
E	2

Routeur B

Destination	Métrique
A	1
B	0
C	2
D	1
E	2

Routeur C

Destination	Métrique
A	1
B	2
C	0
D	1
E	2

Routeur D

Destination	Métrique
A	1
B	1
C	1
D	0
E	1

Routeur E

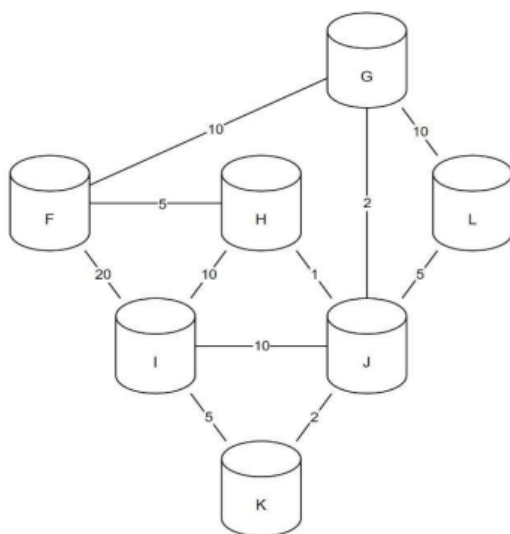
Destination	Métrique
A	2
B	2
C	2
D	1
E	0

- a. Donner la liste des routeurs avec lesquels le routeur A est directement relié.
- b. Représenter graphiquement et de manière sommaire les 5 routeurs ainsi que les liaisons existantes entre ceux-ci.

3. Le protocole OSPF est un protocole de routage qui cherche à minimiser la somme des métriques des liaisons entre routeurs.
- Dans le protocole de routage OSPF le débit des liaisons entre routeurs agit sur la métrique via la relation :  $\text{métrique} = \frac{10^8}{\text{débit}}$  dans laquelle le débit est exprimé en bit par seconde (bps).
- On rappelle qu'un kbps est égal à  $10^3$  bps et qu'un Mbps est égal à  $10^6$  bps.
- Recopier sur votre copie et compléter le tableau suivant :

Débit	100 kbps	500 kbps	?	100 Mbps
Métrique associée	1 000	?	10	1

4. Voici la représentation d'un réseau et la table de routage incomplète du routeur F obtenue avec le protocole OSPF :



Routeur F

Destination	Métrique
F	0
G	8
H	5
I	
J	
K	
L	

Les nombres présents sur les liaisons représentent les coûts des routes avec le protocole OSPF.

- Indiquer le chemin emprunté par un message d'un ordinateur du réseau F à destination d'un ordinateur du réseau I.  
Justifier votre réponse.
- Recopier et compléter la table de routage du routeur F.
- Citer une unique panne qui suffirait à ce que toutes les données des échanges de tout autre réseau à destination du réseau F transitent par le routeur G.  
Expliquer en détail votre réponse.