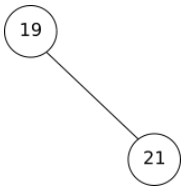
Éléments de correction sujet 02

Exercice 1

1 a
4 feuilles : 12 ; val ; 21 ; 32

1 b

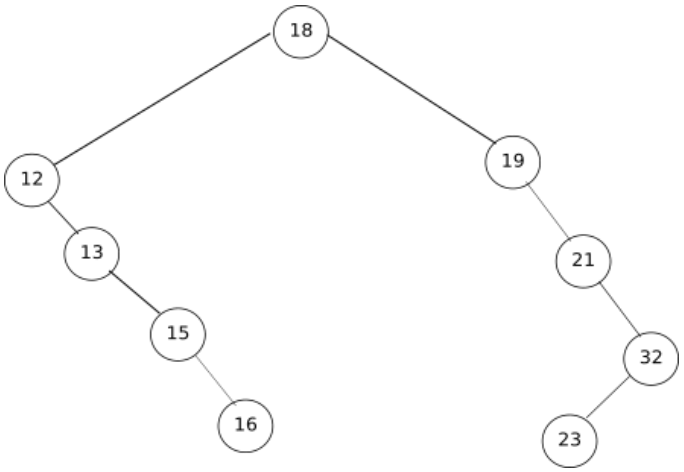


1 c
hauteur = 4 ; taille = 9

1 d
16 ou 17

2 a
infixe : 12 – 13 – 15 – 16 – 18 – 19 – 21 – 23 – 32
suffixe : 12 – 13 – 16 – 15 – 21 – 19 – 32 – 23 – 18

3 a



3 b

```
racine = Noeud(18)
racine.insere_tout([15, 23, 13, 16, 12, 19, 21, 32])
```

il y a d'autres solutions possibles, par exemple :

```
racine = Noeud(18)
racine.insere_tout([15, 13, 12, 16, 23, 19, 21, 32])
```

3 c
bloc 3 – bloc 2 – bloc 1

4

```
 def recherche (self,v):
         n = self
         while n is not None:
             if v < n.v:
                 n = n.ag
             elif v > n.v:
                 n = n.ad
             else:
                 return True
         return False
```

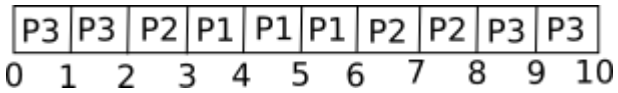
Exercice 2

PARTIE A

1b
2c
3b
4d

PARTIE B

1



2

Il s'agit du scénario 2 car nous nous retrouvons dans la situation où P1 possède R1 et attend R2 avant de pouvoir continuer et P3 possède R2 et attend R1 avant de pouvoir continuer.

PARTIE C

1a

0100 0110 => 46 en hexa => caractère F
0110 0011 => 63 en hexa => caractère c

donc cF

1b

0b 1000 1101 1011 0110

2a

a	b	(a xor b) xor b
0	0	0
0	1	0
1	0	1
1	1	1

2b

On peut remarquer que (a xor b) xor b permet de retrouver a, donc si a correspond au message non chiffré et a xor b correspond au message chiffré, un (a xor b) xor b permet donc de retrouver le message non chiffré. Si on appelle m le message non chiffré, m' le message chiffré et k la clé de chiffrement, un m' xor k permettra de retrouver m.

Exercice 3

1

SGBD => Système de Gestion des Bases de Données

2a

La première requête efface de la table Train le train n°1241 ; la deuxième requête fait donc référence ("Reservation.numT=1241") à une entrée qui n'existe plus dans la base Train ce qui va provoquer une erreur.

2b

Il est impossible d'effectuer une réservation pour un train qui n'existe pas dans la table Train

3a

```
SELECT numT
FROM Train
WHERE destination = 'Lyon'
```

3b

```
INSERT INTO Reservation
(numR, nomClient, prenomClient, prix, numT)
VALUES
(1307, 'Turing', 'Alan', 33,  654)
```

3c

```
UPDATE Train
SET horaireArrivee = 08:11
WHERE numT = 7869
```

4

Cette requête permet de dénombrer les réservations faites au nom de Grace Hopper

5

```
SELECT destination, prix
FROM Train
INNER JOIN Reservation ON Train.numT = Reservation.numT
WHERE nomClient = 'Hopper' AND prenomClient  = 'Grace'
```

Exercice 4

1a

$O(n.\log_2(n))$

1b

L'algorithme de tri par insertion a une complexité en temps dans le pire des cas en $O(n^2)$.
L'algorithme du tri par insertion est moins efficace que l'algorithme de tri fusion.

2
Voici l'affichage obtenu :

[7, 4, 2, 1, 8, 5, 6, 3]
[7, 4, 2, 1]
[7,4]
[2, 1]
[8, 5, 6, 3]
[8, 5]
[6, 3]
résultat renvoyé par la fonction : [1, 2, 3, 4, 5, 6, 7, 8]

3

```
def moitie_droite(L):
    n  = len(L)
    deb = n//2
    tab = []
    for i in range(deb,n):
        tab.append(L[i])
    return tab
```

4

```
def fusion(L1, L2):
    L=[]
    n1 = len(L1)
    n2 = len(L2)
    i1 = 0
    i2 = 0
    while i1<n1 or i2<n2:
        if i1>=n1:
            L.append(L2[i2])
            i2 = i2+1
        elif i2>=n2:
            L.append(L1[i1])
            i1=i1+1
        else :
            e1 = L1[i1]
            e2 = L2[i2]
            if e1 > e2:
                L.append(e2)
                i2 = i2 + 1
            else :
                L.append(e1)
                i1 = i1 + 1
    return L
```

Exercice 5

1a

L'extrait de la table de routage de R1 montre que pour atteindre le réseau L2 (57.37.122.0/24) les paquets doivent être envoyés via l'interface 86.154.10.56. Cette interface fait partie du réseau 86.154.10.0/24. Le routeur R2 fait aussi partie de ce réseau. On peut donc affirmer que depuis R1, les paquets seront dirigés vers R2.

1b

L1 -> R1 -> R2 -> R6 -> L2

2a

L1 -> R1 -> R3 -> R4 -> R6 -> L2

2b

Vu le chemin choisi à la question 2a, seule la ligne R1 sera modifiée (réseau 112.44.65.0 à la place du réseau 86.154.10.0).

3a

C = 10^9 / 10^7 = 100

ATTENTION : la valeur que l'on trouve habituellement (10^8) dans la formule du coût n'est visiblement pas universelle.

3b

la route avec le coût minimum (103) est la suivante : L1 -> R1 -> R2 -> R4 -> R5 -> R6 -> L2

3c

Les tables de routage R2 et R4 seront modifiées.