Henry Letellier
T1

mercredi 31 mars 2021

Numériques et Sciences de l'Informatique (NSI)
DST d'NSI du 31 03 2021
Tiers Temps

NSI
DST d'NSI du 31 03 2021

Note :	Appréciations :	Signature :
_____		

1) A) Cet arbre a quatre feuilles, ses quatre feuilles sont.
   - feuille 1 : « 12 »
   - feuille 2 : « Val »
   - feuille 3 : « 21 »
   - feuille 4 : « 32 »

   B) Le sous-arbre gauche du nœud 23 est : 19 ,l'enfant de droite de la branche 19 est la feuille numéro 21.
   C) L'arbre a une hauteur de 3 et une taille de 6.
   D) Les valeurs entières de val pour cet arbre binaire sont : 16 (soit 15+1), 17 (soit 15+2), 18 (soit 15+3)

2) val=16
   a) Les valeurs d'affichage des nœuds pour le parcourt infixe de l'arbre sont : 15,12,13,16,18,23,21,19,32
   b) Les valeurs d'affichage des nœuds pour le parcourt suffixe de l'arbre sont : 12, 13, 16, 15, 21,19,32,23,18

3)

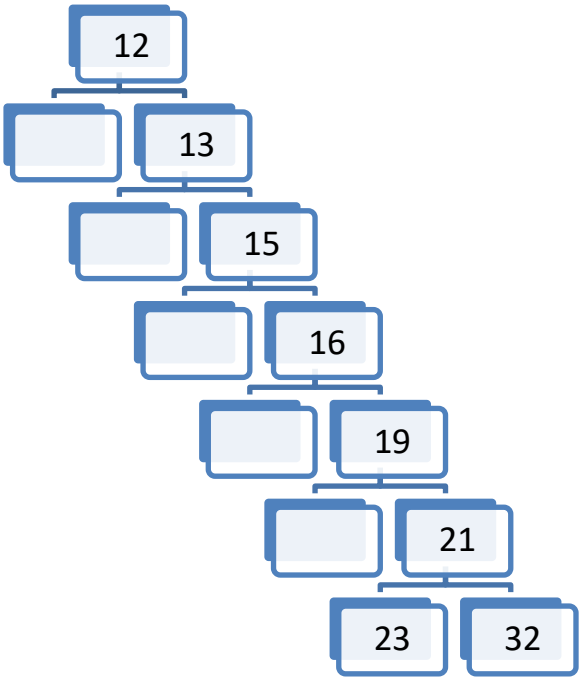
```
1)  class Noeud():
2)      def __init__(self,v):
3)          self.ag=None
4)          self.ad=None
5)          self.v=v
6)
7)      def insere(self, v):
8)          n=self
9)          est_insere=False
10)         while not est_insere:
11)             if v==n.v:
12)                 est_insere=True
13)             elif v<n.v:
14)                 if n.ag!=None:
15)                     n=n.ag
16)                 else:
17)                     n.ag=Noeud(v)
18)                     est_insere=True
19)             else:
20)                 if n.ad!=None:
21)                     n=n.ad
22)                 else:
23)                     n.ad=Noeud(v)
24)                     est_insere=True
25)     def insere_tout(self, vals):
26)         for v in vals:
27)             self.insere(v)
```

Bloc 1

Bloc 2

Bloc 3

a) Suite à l'instruction :
racine=Nœud(18)
racine.insere_tout([12,13,15,16,19,21,32,23]) sera : (Les boites vides servent juste à la disposition, si l'arbre venait à être transcrit sur du papier, elles disparaitraient).



b) Si le nombre suivant est supérieur au précédent, insérer à Droite sinon, insérer à gauche.

c) B1 = Bloc 1, B2 = Bloc 2, B3 = Bloc 3
L'ordre d'exécution des blocs est le suivant : B3, B3, B2, B3, B1, B1, B3, B1, B1, B2, B2, B2, B1, B1, B3, B1, B1, B1.

4) Le code pour la valeur de recherche est :

```
1) def recherche(self,v):
2)         found=trouver_nombre(self,v)#trouver nombre est
   une fonction définie qui renvoie NULL si non trouvé, ou le
   nombre si trouvé.
3)         if found.isnumeric():return True
4)         else: return False
5)
```

Exercice 2 :
Partie A :

1) B
2) C
3) B
4) D

Partie B :

1)

P3	P2	P1	P1	P1	P2	P2	P3	P3	P3
0	1	2	3	4	5	6	7	8	9	10

2)

Le Scénario 1 provoquent un interblocage car les trois ressources disponibles sont utilisées pour acquérir une et même ressource, la ressource 1. Dans les autres cas, il y a au moins une ressource de disponible pour continuer à faire d'autres tâches.

Partie C :

1) m=0b 0110 0011 0100 0110

Conversion du binaire en base de 10 :
Longueur du binaire : 16
Final= $1*2^1+1*2^2+1*2^6+1*2^7+1*2^9+1*2^{13}+1*2^{14}$ = 2+4+64+128+512+8192+16384 = 25286

a) .
b) Clé= 0b1110 1110 1111 0000

Message :	0	1	1	0		0	0	1	1		0	1	0	0		0	1	1	0
Clé :	1	1	1	0		1	1	1	0		1	1	1	1		0	0	0	0
Résultat :	1	0	0	0		1	1	0	1		1	0	1	1		0	1	1	0

2) A)

a	b	a XOR b	(a XOR b) XOR b	Raisonnement
0	0	0	0	(0XOR0)=0XOR0=0
0	1	1	0	(0XOR1)=1XOR1=0
1	0	1	1	(1XOR0)=1XOR0=1
1	1	0	1	(1XOR1)=0XOR1=1

b) Bob devra utiliser la table a XOR b pour déchiffrer le message.

Exercice 3 :

1) Le nom générique donné au logiciel qui assure entre autre la persistance des données, l'efficacité du traitement des requêtes et la sécurisation des accès pour les bases de données est le SGBD.
2) .
   a) .

```
DELETE FROM Train WHERE numT = 1241 ;
DELETE FROM Reservation WHERE numT = 1241 ;
```

Cette requête posera un problème car la colonne numT dans la table Reservation car la variable numT est une clé étrangère à la table Reservation, ainsi, en supprimant d'abord la variable numT de la table Train on la supprime aussi de la table Reservation, donc quand on souhaite ensuite la supprimer de la table Reservation, comme cette dernière n'y est plus, SQL ne pourras donc pas effectuer l'opération.

b) Le cas serais de vouloir insérer uniquement des informations dans la table Reservation sans aussi remplir la table Train car ces deux dernières vas chercher la variable numT dans le tableau Train et ne peut donc pas la créer.

3)

a) La requête pour afficher tous les numéros de trains dont la destination est « Lyon » est :

```
SELECT FROM Train WHERE destination = « Lyon » ;
```

b) La requête pour ajouter une réservation n°1307de 33€ pour M. Alan Turing dans le train n°654 est :

```
INSERT INTO VALUES (numR = 1307, prix= « 33€ », nomClient= « Turing », prenomClient= « Alan », #numT=654) WHERE Reservation;
```

c) La requête pour modifier l'horraire d'arrivé du train n°7869 est :

```
SELECT FROM Train WHERE numT = 7869 UPDATE horaireArrivee = « 08 :11 »;
```

4)

```
SELECT COUNT (*) FROM Reservation WHERE nomClient= « Hopper » AND prenomClient= « Grace » ;
```

Cette requête, permet de déterminer le nombre de personnes du nom de Hopper et de prénom Grace dans toute la table Reservation.

5) La requête qui renvoie les destinations et les prix des réservations effectués par Grace Hopper est :

```
SELECT (* destination AND prix AND nomClient= « Hopper » AND prenomClient= « Grace ») FROM Train AND Reservation ;
```

Exercice 5 :

De/vers=vers/de	IP/base
L1 → R1	192.168.1.0/24
R1 → R3	112.44.65.0/24
R1→ R2	86.154.10.0/24
R3 → R4	62.34.2.0/24
R3 → R2	176.139.8.0/24
R2 → R4	212.194.171.0/24
R2 → R5	10.94.75.0/24
R2 → R6	37.49.236.0/24
R4 → R5	87.3.5.0/24
R4 → R6	94.23.122.0/24
R6 → R5	218.32.15.0/24
R6 → L2	54.37.122.0/24

1) De L1 à L2
   A) Le routeur enverras sont paquet à R2 car c'est le premier dans sa liste.
   B) Le paquet fera, si R1 envoie à R2, et que chaque routeur prend le routeur avec le chiffe le plus grand après sont émetteur de sa table , L1→R1→R2→R3→R4→R5→R6→L2
2) La liaison entre R1 et R2 est rompue.
   a. Le paquet pourras suivre le chemin L1→R1→R3→R4→R6→L2.
   b. On supprimera la ligne R2 car ce dernier est « down ».