

Nom : \_\_\_\_\_

Code permanent : \_\_\_\_\_

Numéro de place : \_\_\_\_\_

Directives pédagogiques :

- Inscrivez votre nom, prénom, code permanent et numéro de votre place.
- Lisez attentivement toutes les questions et **répondez directement sur le questionnaire.**
- Seule l'utilisation d'un crayon est permise, **aucune documentation, calculatrice, téléphone cellulaire, ordinateur, ou autre objet permis.**
- Cet examen contient 10 questions pour 150 points au total.
- Il y a 15 points en bonus.
- Cet examen contient 21 pages, incluant 4 pages détachables à la fin pour vos brouillons.
- **Écrivez lisiblement et détaillez vos réponses.**
- Vous avez 160 minutes pour compléter cet examen.

BONNE CHANCE !

1	/ 10
2	/ 15
3	/ 15
4	/ 10
5	/ 20
6	/ 15
7	/ 10
8	/ 20
9	/ 25
10	/ 25
Total	/150

1. (10) Montrez que  $\sum_{i=1}^n i^2$  est  $O(n^3)$ .

**Deux solutions possibles sont :**

- 1) une preuve par induction visuelle; ou,**
- 2) borner la somme par une intégrale.**

2. (15) Soit A une liste de taille  $n \geq 2$  et qui contient les entiers de 1 à  $n-1$ , incluant une et une seule répétition. Décrivez un algorithme rapide,  $O(n)$ , pour trouver l'entier répété dans A.

**Il ne faut surtout pas trier le tableau A !**

```
def missing(A):  
    found = [False] * len(A)  
    for val in A:  
        if found[val]: return val  
        else:  
            found[val] = True
```

3. (15) Supposez qu'on dépose trois valeurs dans un pile S dans un ordre aléatoire. Donnez un code sans boucle ni récursivité qui n'utilise qu'une et une seule comparaison et une et une seule variable, x, tel que la valeur de x à la fin de l'exécution de votre code contient la plus grande valeur des trois avec probabilité de 2/3. Dites pourquoi votre méthode est correcte.

```
x = S.pop()  
if x < S.top():  
    x = S.pop()
```

4. (10) Un aéroport veut développer un système de simulation du trafic aérien pour gérer ses événements d'arrivées et de départs. Chaque événement est étiqueté de son temps, soit le temps du moment où l'événement survient. Le programme de simulation doit faire les deux opérations fondamentales suivantes:
1. Insérer un événement futur à un temps donné.
  2. Extraire le prochain événement.

Quelle structure de données devrait être utilisée pour implanter ces opérations et pourquoi?

**La meilleure structure de données pour ce programme de simulation est la queue avec priorités. La queue va gérer tous les événements et conserver l'ordre tel que le prochain événement à survenir (avec le temps le plus petit) peut être observé ou retiré de manière efficace (avec min et remove).**

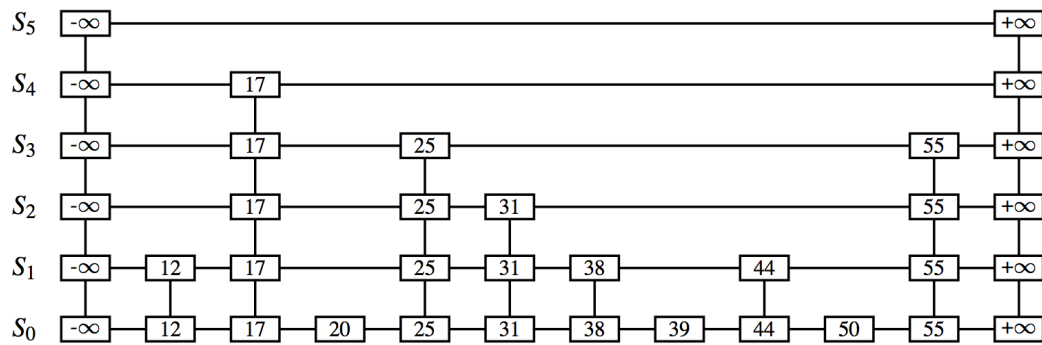
5. (20) Dessinez la table de hachage qui résulte de l'utilisation de la fonction de hachage  $h(i) = (3i+5) \bmod 11$ , pour insérer les clés suivantes : 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 et 5.
- a) (10) Si les collisions sont résolues par un chaînage externe.

0	1	2	3	4	5	6	7	8	9	10
13	94				44			12	16	20
	39				88			23	5	
					11					

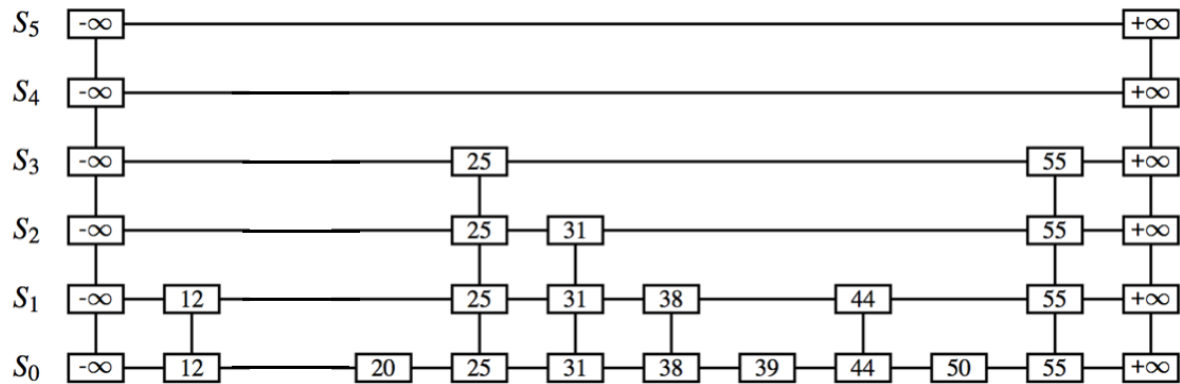
- b) (10) Si les collisions sont résolues par sondage linéaire.

0	1	2	3	4	5	6	7	8	9	10
13	94	39	16	5	44	88	11	12	23	20

6. (10) Considérez la « skip list » suivante :

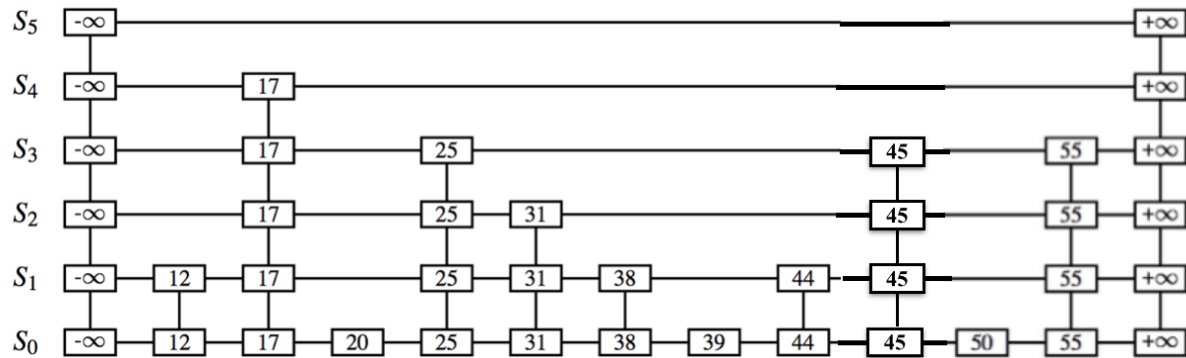


a) (5) Dessinez la skip list résultante après l'opération `del S[17]`.



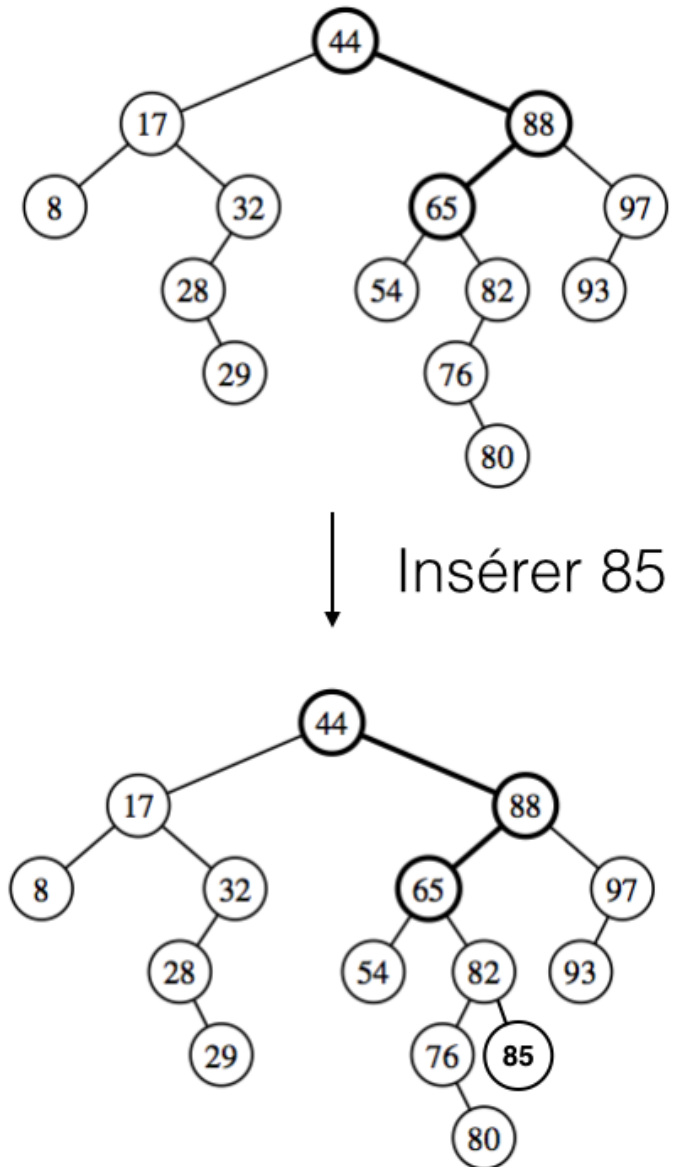


- b) (5) À partir de la « skip list » ci-haut (celle de départ et non pas celle que vous avez obtenue en (a) ), dessinez la « skip list » résultante après l'opération  $S[45] = 'x'$ , si le hasard détermine d'incrémenter la hauteur de sa tour 3 fois.

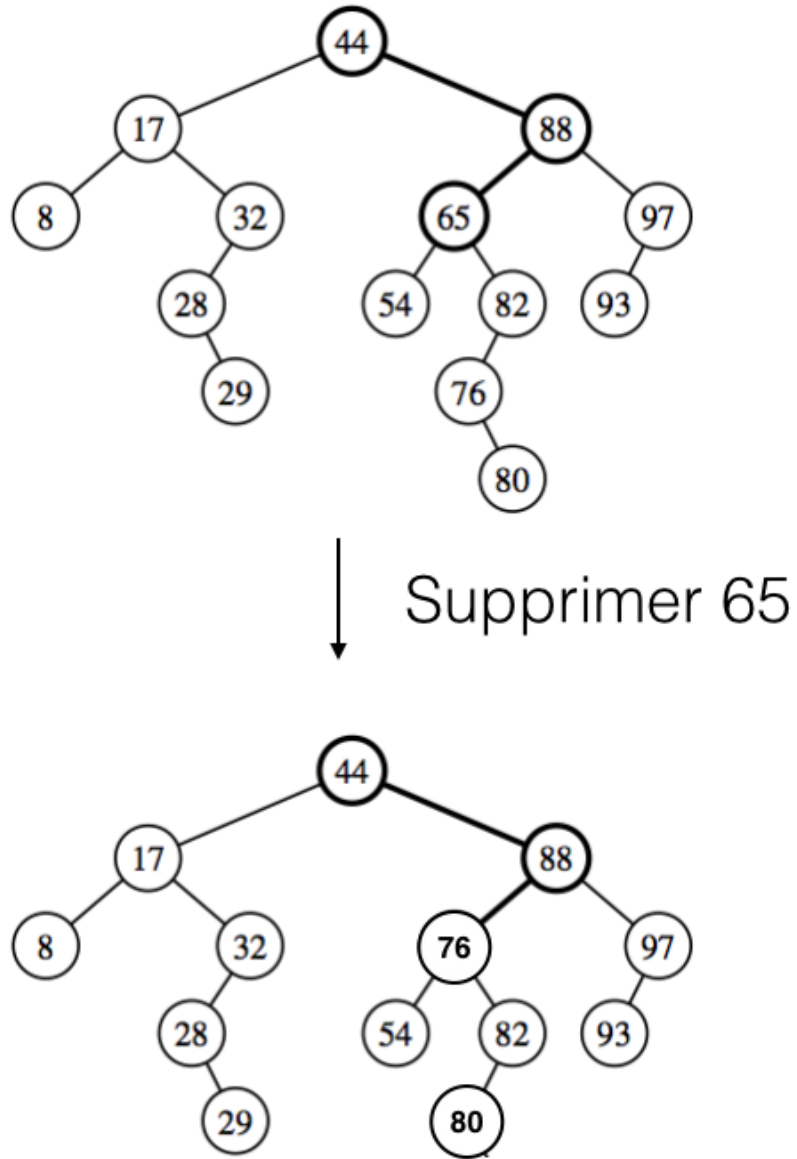


7. (10) Effectuez les opérations suivantes sur les arbres binaires de recherche (ABR).

a) (5) Insertion d'une clé.

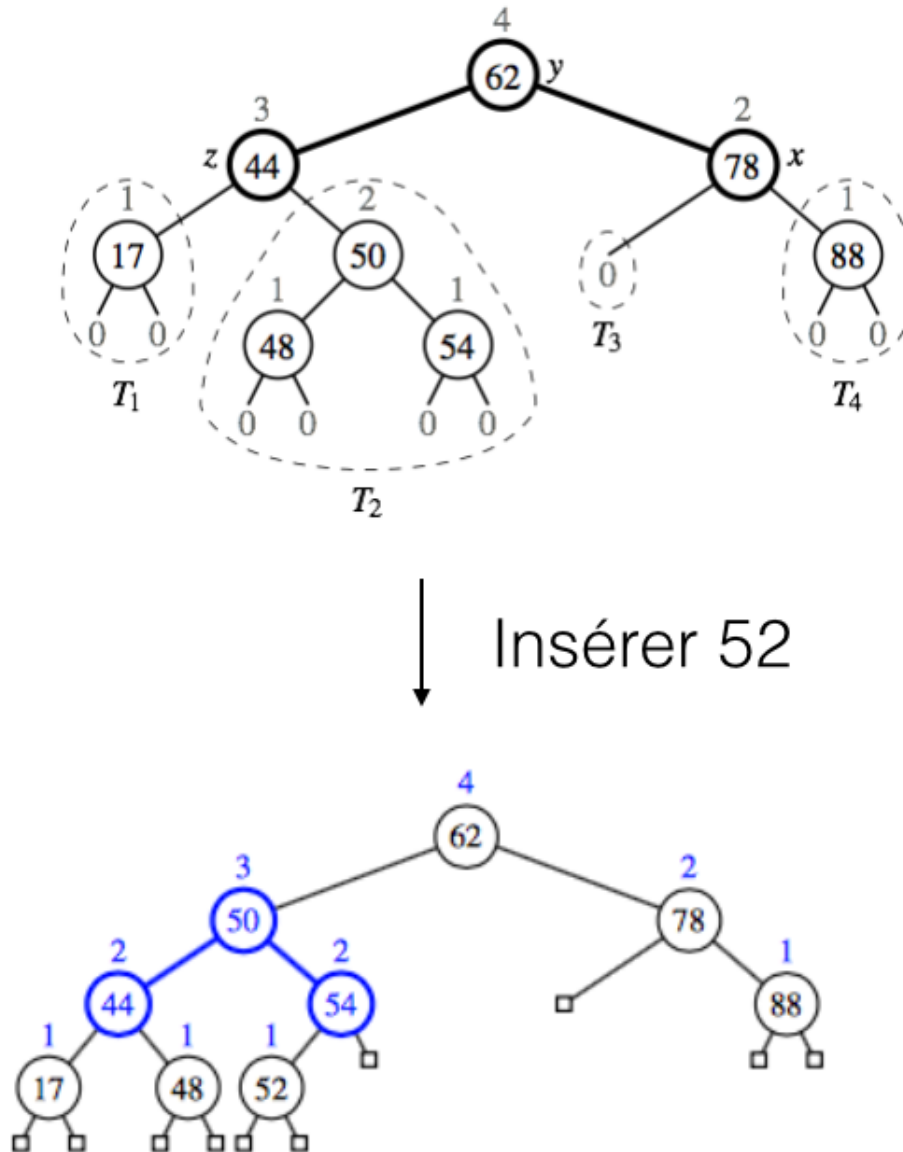


b) (5) Suppression d'une clé.

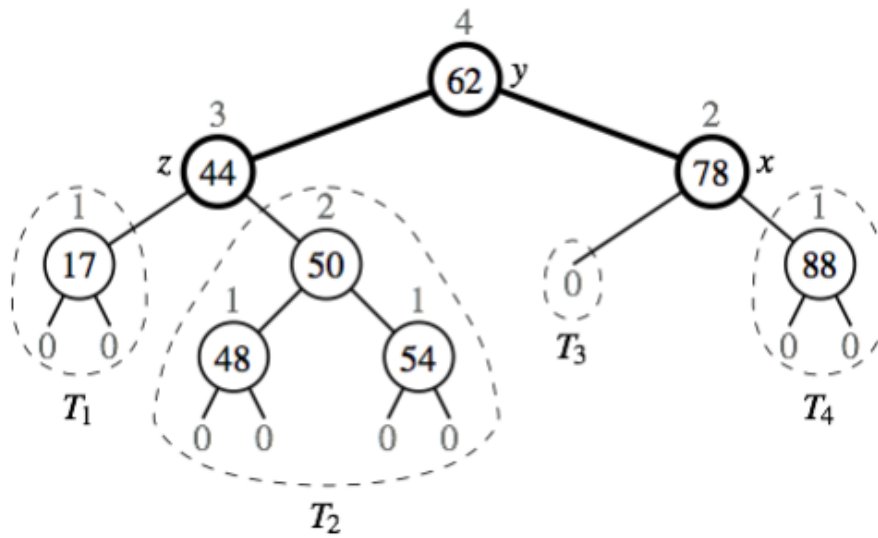


8. (20) Effectuez les opérations d'insertion et de suppression dans les arbres AVL. Dessinez l'arbre AVL résultant et indiquez la hauteur de chaque noeud.

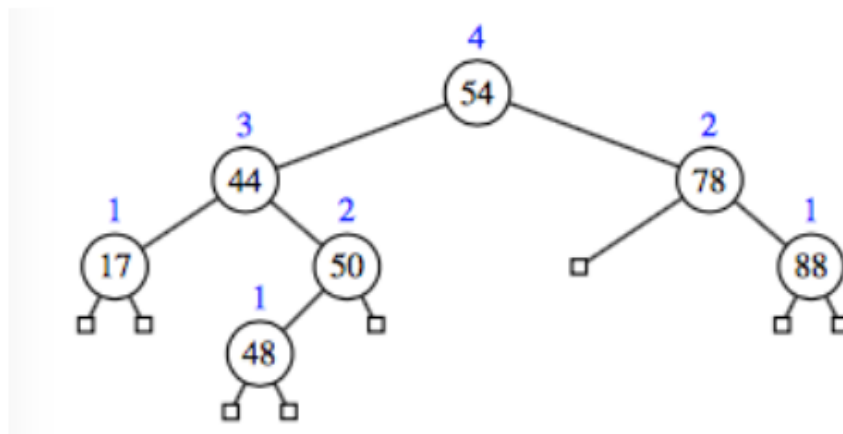
- a) (10) Insertion d'une clé.



b) (10) Suppression d'une clé.

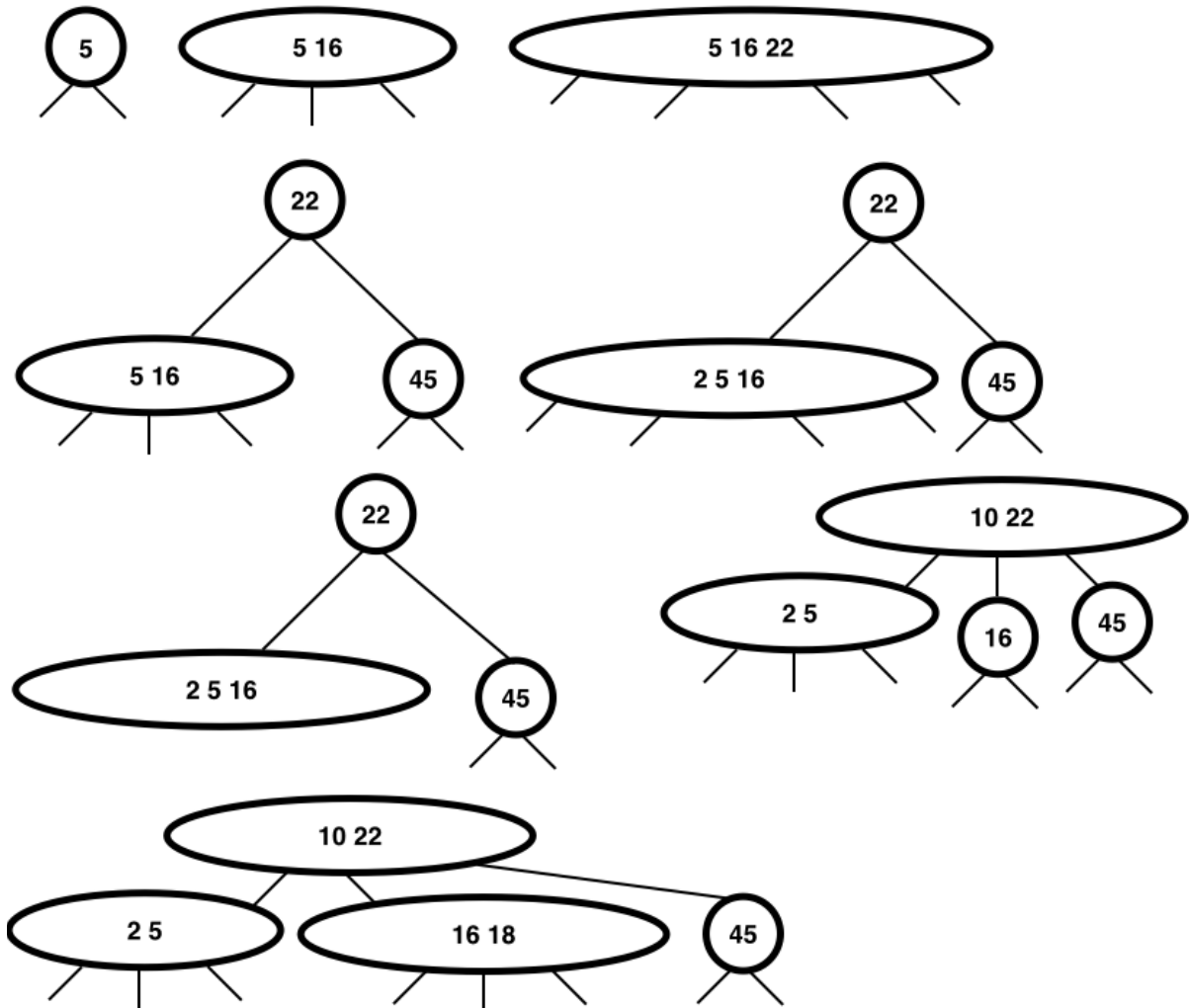


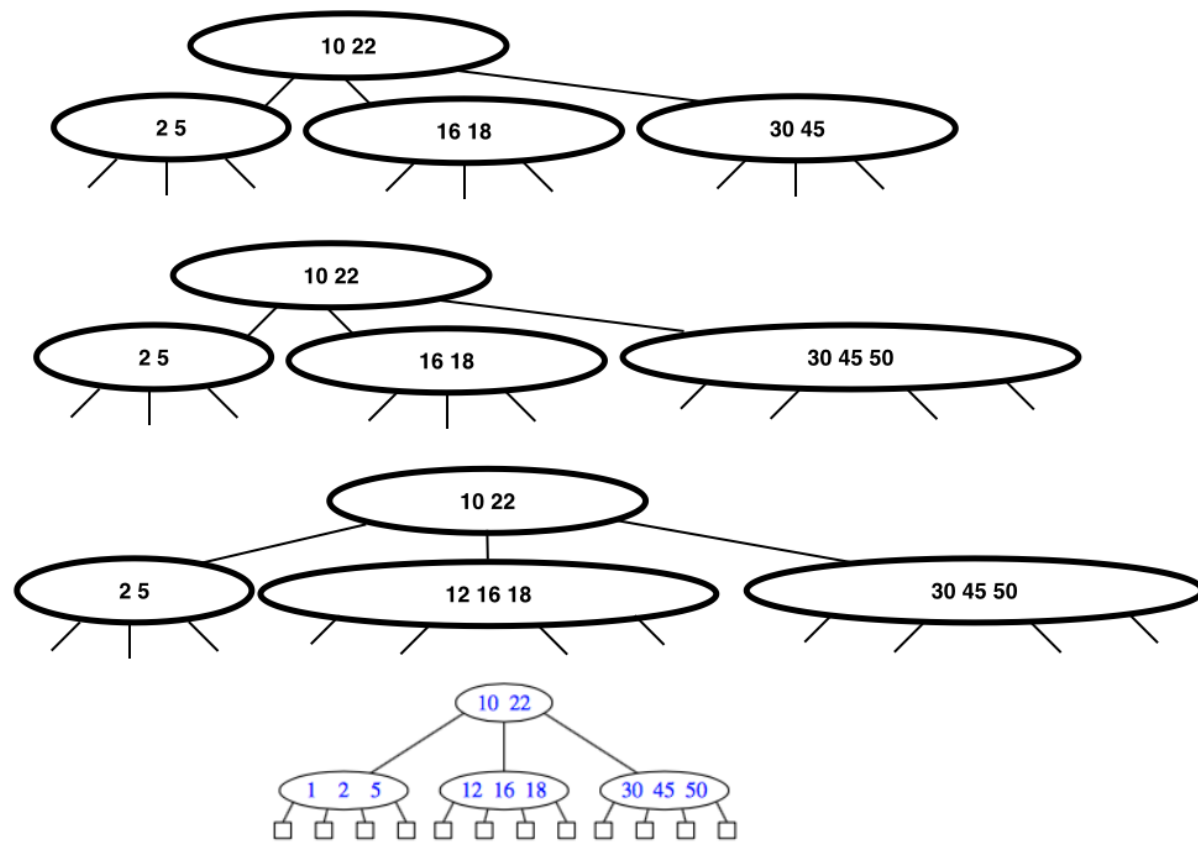
Supprimer 62



**PS. On peut aussi remplacer par 78 mais il faudra rebalancer l'AVL correctement.**

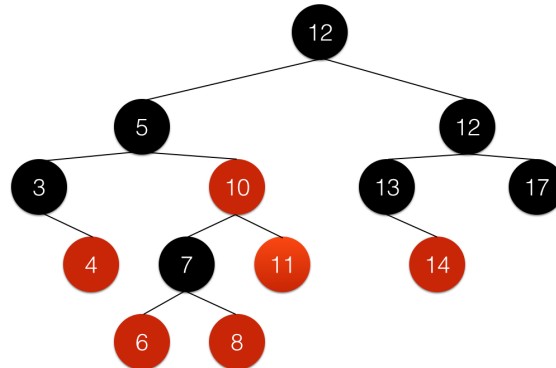
9. (25) Considérez l'insertion des clés suivantes : 5, 16, 22, 45, 2, 10, 18, 30, 50, 12 et 1. Dessinez les arbres obtenus après chaque insertion si ces clés sont insérées dans cet ordre dans un arbre (2,4) initialement vide.





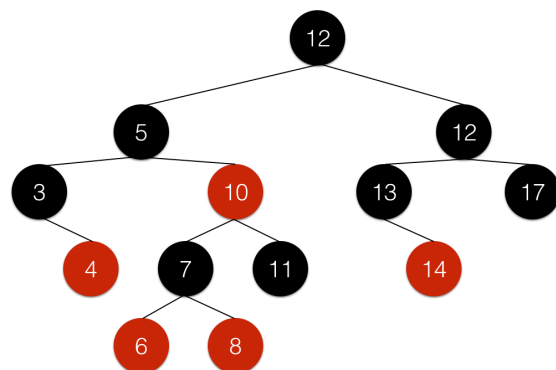
10. (25) Dites si les arbres rouge-noir suivants sont valides. Dans le cas où un arbre n'est pas valide indiquez les propriétés qui ne sont pas respectées.

a) (5)



**Non valide. D'abord, les enfants du noeud rouge 10 devraient être noirs alors que 11 est rouge. Ensuite, les noeuds de un ou zéro enfant n'ont pas la même profondeur noire (e.g. 13 a une profondeur noire de 3 alors que 11 a une profondeur noire de 2).**

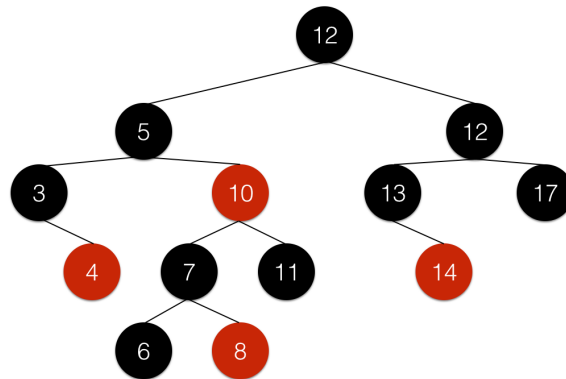
b) (5)



**Valide.**

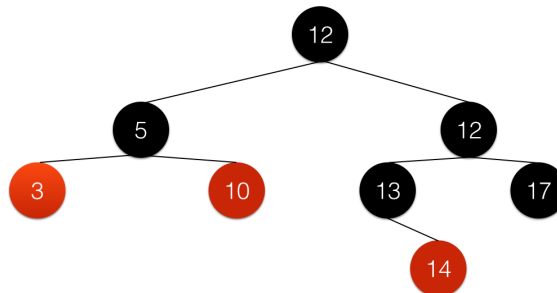


c) (5)



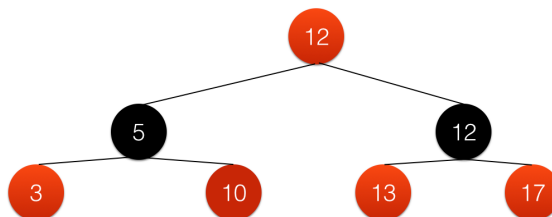
**Non valide. Profondeur noire de 6 est 4 alors que pour les autres noeuds de 1 ou zéro enfant elle est de 3.**

d) (5)



**Non valide. Profondeur noire de 3 est 2 alors que pour 13, 14 et 17 elle est de 3.**

e) (5)



**Non valide. La racine est rouge.**