

Algorithmique

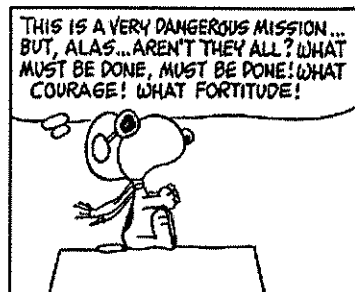
Contrôle n° 3 (C3)

INFO-SPÉ - S3
EPITA

23 octobre 2023

Consignes (à lire) :

- Vous devez répondre sur les feuilles de réponses prévues à cet effet.
 - Aucune autre feuille ne sera ramassée (gardez vos brouillons pour vous).
 - Répondez dans les espaces prévus, les réponses en dehors ne seront pas corrigées : utilisez des brouillons !
 - Ne séparez pas les feuilles à moins de pouvoir les ré-agrafer pour les rendre.
 - Aucune réponse au crayon de papier ne sera corrigée.
- La présentation est notée en moins, c'est à dire que vous êtes noté sur 20 et que les points de présentation (2 au maximum) sont retirés de cette note.
- Le code :
 - Tout code doit être écrit dans le langage Python (pas de C, CAML, ALGO ou autre).
 - **Tout code Python non indenté ne sera pas corrigé.**
 - Tout ce dont vous avez besoin (classes, fonctions, méthodes) est indiqué en **annexe** !
 - Vous pouvez également écrire vos propres fonctions, dans ce cas elles doivent être documentées (on doit savoir ce qu'elles font).
Dans tous les cas, la dernière fonction écrite doit être celle qui répond à la question.
 - Comme d'habitude l'**optimisation** est notée. Si vous écrivez des fonctions non optimisées, vous serez notés sur moins de points.¹
- Durée : 2h



1. Des fois, il vaut mieux peu de points que pas de point.

Exercice 1 (Haches et graphes... – 6 points)

1. Donnez une méthode de hachage indirect.
2. Quel problème apparait avec le hachage coalescent ?
3. Qu'est-ce qu'une collision primaire ?
4. Qu'est-ce que l'ordre d'un graphe non orienté ?
5. Qu'est-ce qu'un sommet isolé ?

Soit le graphe $G = \langle S, A \rangle$ orienté avec :

$S = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

et $A = \{(1, 2), (1, 6), (2, 3), (2, 5), (3, 1), (3, 4), (3, 5), (4, 5), (4, 8), (6, 2), (6, 5), (7, 5), (7, 6), (7, 8), (8, 5), (8, 9), (9, 4), (9, 7)\}$

6. Remplir le tableau des demi-degrés extérieurs des sommets de G .
7. Le graphe G est-il fortement connexe ?
8. Si NON, combien possède-t-il de composantes fortement connexes ?
9. S'ils existent, quels sont les sommets de G de degré différent de 4 ?
10. S'ils existent, quels sont les sommets de G de demi-degré intérieur égal à 0 ?

Exercice 2 (Level from x – 5 points)

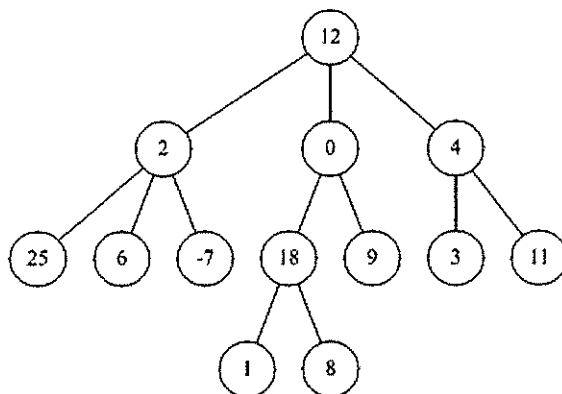


FIGURE 1 – Arbre général T4

Écrire la fonction `keys_after(T:Tree, x:int)` qui retourne la liste des clés dans l'arbre général T (implémentation "classique") se trouvant après la première valeur x trouvée sur le même niveau que x.

Exemples d'applications avec T4 l'arbre de la figure 2 :

```
1 >>> keys_after(T4, 12)
2 []
3 >>> keys_after(T4, 2)
4 [0, 4]
5 >>> keys_after(T4, 0)
6 [4]
7 >>> keys_after(T4, 4)
8 []
9 >>> keys_after(T4, 6)
10 [-7, 18, 9, 3, 11]
11 >>> keys_after(T4, 42)
12 []
```

Exercice 3 (Average subtrees – 6 points)

Dans un arbre général, on cherche à savoir si aucun nœud n'a un sous-arbre dont la moyenne des clés est strictement supérieure à la clé de ce nœud.

Par exemple, l'arbre de la figure 2 ne respecte pas la propriété : la moyenne des clés du premier sous-arbre du nœud contenant 4 est supérieure à 4.

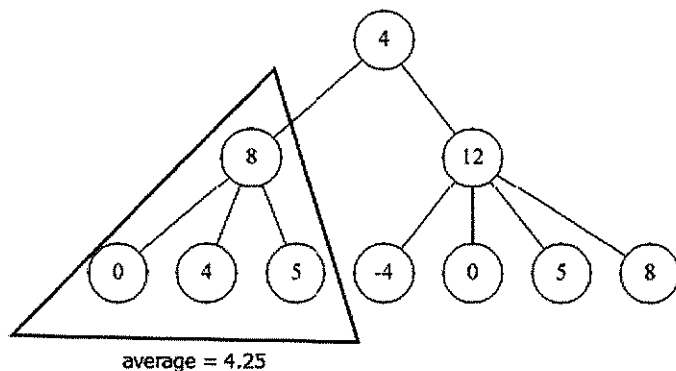


FIGURE 2 – Propriété fausse

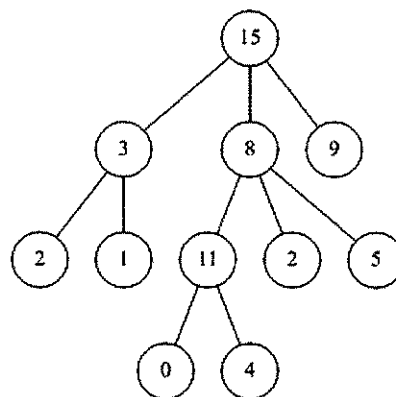


FIGURE 3 – Propriété vraie

Par contre l'arbre de la figure 3 respecte bien la propriété :

- Le nœud contenant 15 :
 - la moyenne des clés du premier sous-arbre est 2
 - la moyenne des clés du deuxième est $5 = (8 + 11 + 2 + 5 + 0 + 4) / 6$
- Nœud contenant 8 : la moyenne des clés du premier sous-arbre est 5, inférieure à 8
- Tous les autres sous-arbres sont réduits à une feuille dont la clé est inférieure à celle du père.

Écrire la fonction `average_subtrees(B:TreeAsBin)` qui vérifie si l'arbre B en implémentation premier fils-frère droit respecte la propriété.

Exercice 4 (B-arbre : insertions et suppression – 3 points)

Pour chaque question, utiliser le principe "à la descente" (principe de précaution) vu en td (hors bonus).

1. Dessiner l'arbre après insertions successives des valeurs 21, 42 et 8 dans l'arbre de la figure 4.

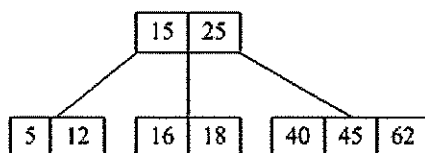


FIGURE 4 – B-arbre B1 pour insertions, degré 2

2. Dessiner l'arbre après suppression de la valeur 80 dans l'arbre de la figure 5.

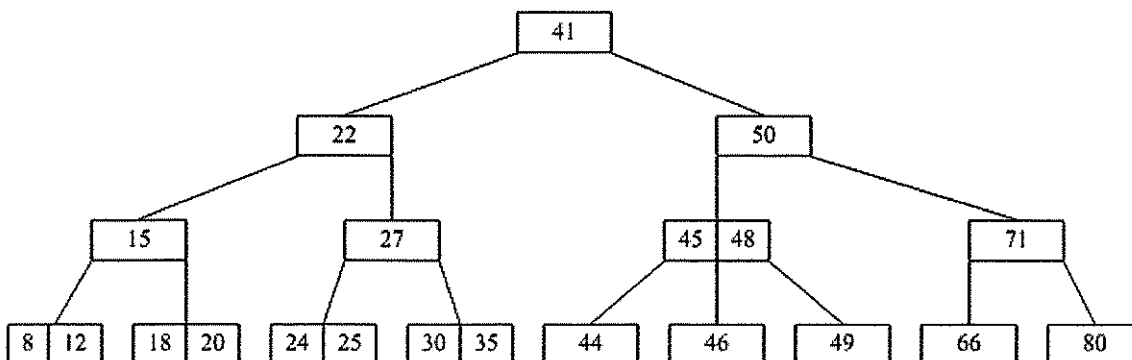


FIGURE 5 – B-arbre B2 pour suppression, degré 2

Annexes

Les arbres généraux

Les arbres (généraux) manipulés ici sont les mêmes qu'en td.

Implémentation classique

- T : classe Tree
- T.key
- T.children : listes des fils ([] pour les feuilles)
- T.nbchildren = len(T.children)

Implémentation *premier fils - frère droit*

- B : classe TreeAsBin
- B.key
- B.child : le premier fils
- B.sibling : le frère droit

Autres fonctions et méthodes autorisées

Comme d'habitude : len, range, min, max, abs, append.

Les méthodes de la classe Queue, que l'on suppose importée :

- Queue() retourne une nouvelle file ;
- q.enqueue(e) enfile e dans q ;
- q.dequeue() supprime et retourne le premier élément de q ;
- q.isempty() teste si q est vide.

Vos fonctions

Vous pouvez également écrire vos propres fonctions à condition qu'elles soient documentées : donnez leurs spécifications (on doit savoir ce qu'elles font).

Dans tous les cas, la dernière fonction doit être celle qui répond à la question.