Polynésie – 2023 – sujet2 - Correction

Exercice 1 (4 points)

1.

1.a. Justifier que cet arbre est un arbre binaire.

Un arbre binaire est un arbre dans lequel chaque nœud possède **au plus deux fils** : un fils gauche et un fils droit.

Ici, chaque nœud possède :

- soit zéro fils (feuille),
- soit un fils,
- soit deux fils,
- mais jamais plus de deux.

👉 *Commentaire : La définition d'un arbre binaire repose uniquement sur le nombre maximal de fils par nœud.*

1.b. Indiquer si l'arbre est un arbre binaire de recherche (ABR).

Un arbre binaire de recherche vérifie la propriété :

- toutes les valeurs du sous-arbre gauche sont strictement inférieures à la valeur du nœud ;
- toutes les valeurs du sous-arbre droit sont strictement supérieures.

Si cette propriété est respectée pour chaque nœud, alors l'arbre est un ABR.

👉 *Commentaire : Pour justifier correctement, il faut vérifier la propriété pour tous les nœuds et pas seulement pour la racine.*

2.

2.a. Compléter l'assertion

L'assertion doit vérifier :

- que mini est un entier ;
- que maxi est un entier ;
- que mini ≤ maxi.

Ligne complétée :

```
assert isinstance(mini, int) and isinstance(maxi, int) and mini <= maxi
```

👉 *Commentaire : Une assertion permet de vérifier des préconditions d'appel de fonction.*

2.b. Appels récursifs pour construire(0,8)

Décomposition :

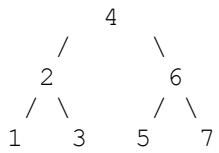
```
construire(0,8)
├── construire(0,4)
│   ├── construire(0,2)
│   └── construire(2,4)
└── construire(4,8)
    ├── construire(4,6)
    └── construire(6,8)
```

Puis chaque appel où maxi - mini ≤ 2 devient un cas de base.

👉 *Commentaire : On observe une division de l'intervalle par 2 à chaque étape : logique de type diviser-pour-régner.*

2.c. Arbre renvoyé par construire(0,8)

L'arbre obtenu est :



👉 *Commentaire : On obtient un arbre parfaitement équilibré.*

2.d. Arbre renvoyé par construire(0,3)

👉 *Commentaire : Même principe mais sur un intervalle plus petit.*

2.e. Parcours infixe de l'arbre 2.c

Parcours infixe = gauche → racine → droite.

Résultat :

```
1, 2, 3, 4, 5, 6, 7
```

👉 *Commentaire : Un parcours infixe d'un ABR donne toujours les valeurs dans l'ordre croissant.*

Cela prouve que l'arbre est un ABR.

2.f. Compléter la fonction maximum

```python
def maximum(abr):
    if abr is None:
        return None
    elif abr.droit is None:
        return abr.valeur
    else:
        return maximum(abr.droit)
```

👉 *Commentaire : Dans un ABR, la valeur maximale se trouve toujours dans le nœud le plus à droite.*

3.

3.a. Résultats de mystere

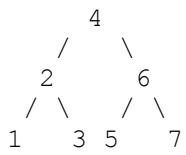
La fonction ajoute chaque valeur visitée dans la liste.

Elle descend :

- à gauche si x < valeur
- à droite si x > valeur

Elle retourne la liste des valeurs parcourues.

Supposons que abr_7_noeuds soit :



mystere(abr_7_noeuds, 5, [])

Chemin : 4 → 6 → 5

Résultat :

```
[4, 6, 5]
```

mystere(abr_7_noeuds, 6, [])

Chemin : 4 → 6

Résultat :

```
[4, 6]
```

mystere(abr_7_noeuds, 2, [])

Chemin : 4 → 2

Résultat :

```
[4, 2]
```

👉 *Commentaire : La liste correspond au chemin suivi dans l'arbre.*

3.b. Rôle de la fonction mystere

Cette fonction :

- recherche la valeur x dans un arbre binaire de recherche ;
- renvoie le chemin parcouru depuis la racine jusqu'au nœud recherché.

👉 *Commentaire : Il s'agit d'une recherche dans un ABR avec mémorisation du chemin.*
