
TD 5 - Arbres binaires de recherche

Inès de Courchelle, Elisabeth Ranisavljevic



2023-2024

**Objectifs :**

- Connaître les arbres binaires
- Utiliser des algos spécifiques
- Comprendre les mécaniques de parcours d'un arbre binaire

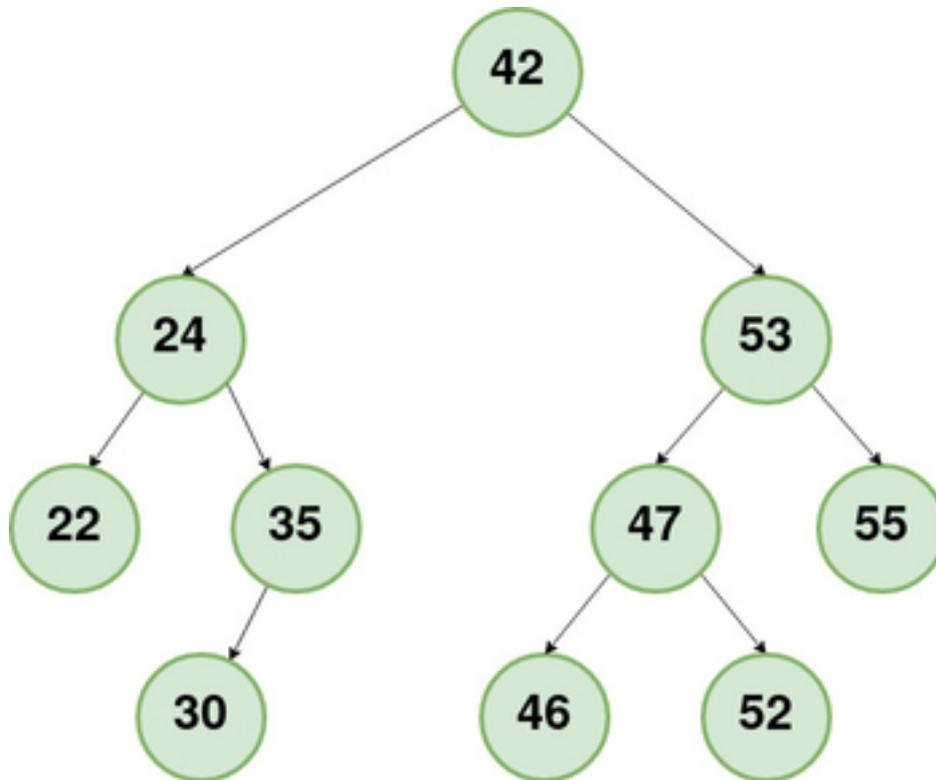
Durée 1h30**Format** papier**Attention** Tous les exercices ne seront pas corrigés en cours. On vous en laisse pour vos révisions !

Arbres binaires de recherche

Exercice 1 : Le commencement

1. Rappelez la définition d'un arbre binaire de recherche

Nous considérons l'arbre suivant :



2. Nous souhaitons rajouter 37 dans cet arbre. Dessinez le nouvel arbre :



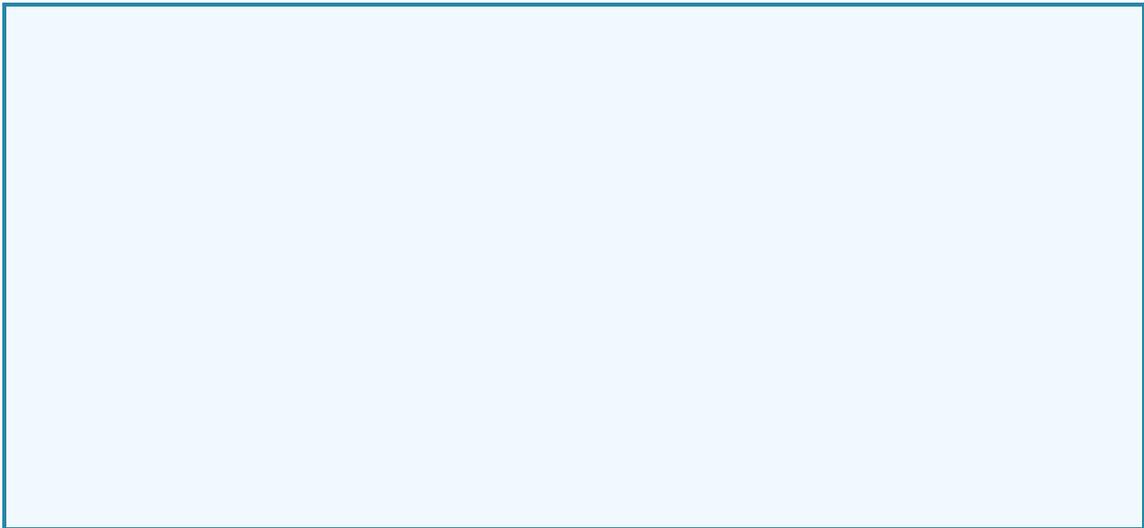
3. Nous souhaitons supprimer 47 dans cet arbre. Dessinez le nouvel arbre :



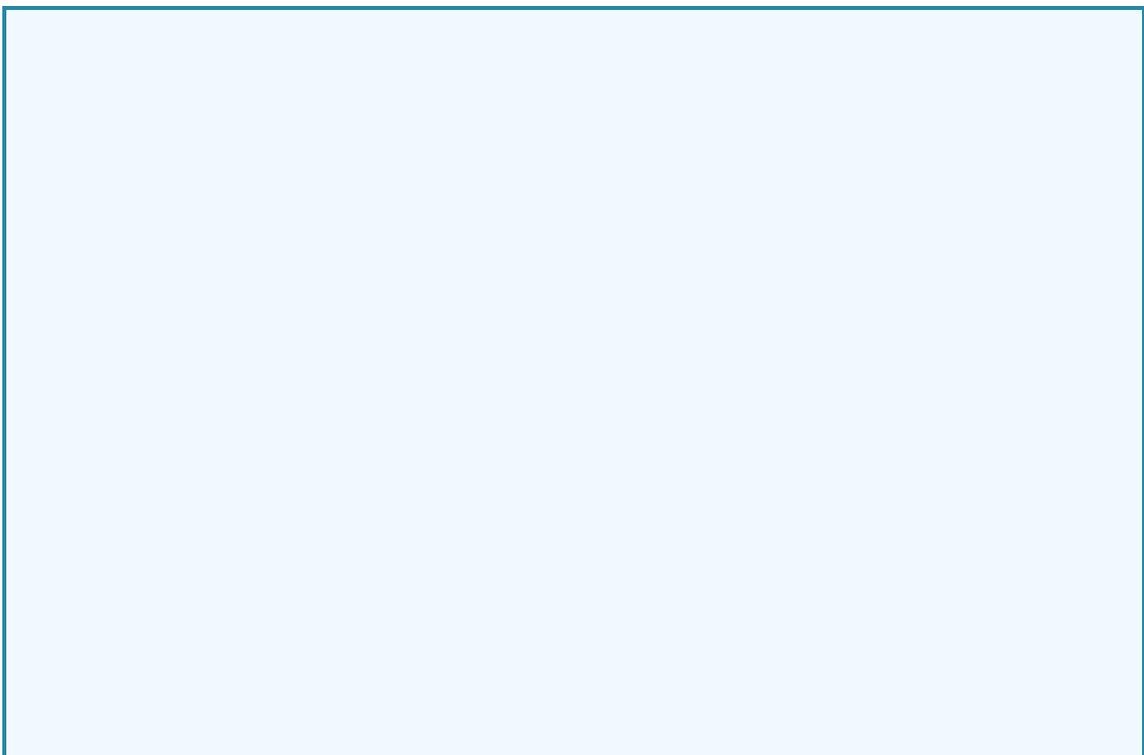
Exercice 2 : Un peu plus loin

Nous considérons l'ensemble des entiers suivants : [1 4 5 10 16 17 21] Dessiner des arbres binaires de recherche de cet ensemble d'entiers

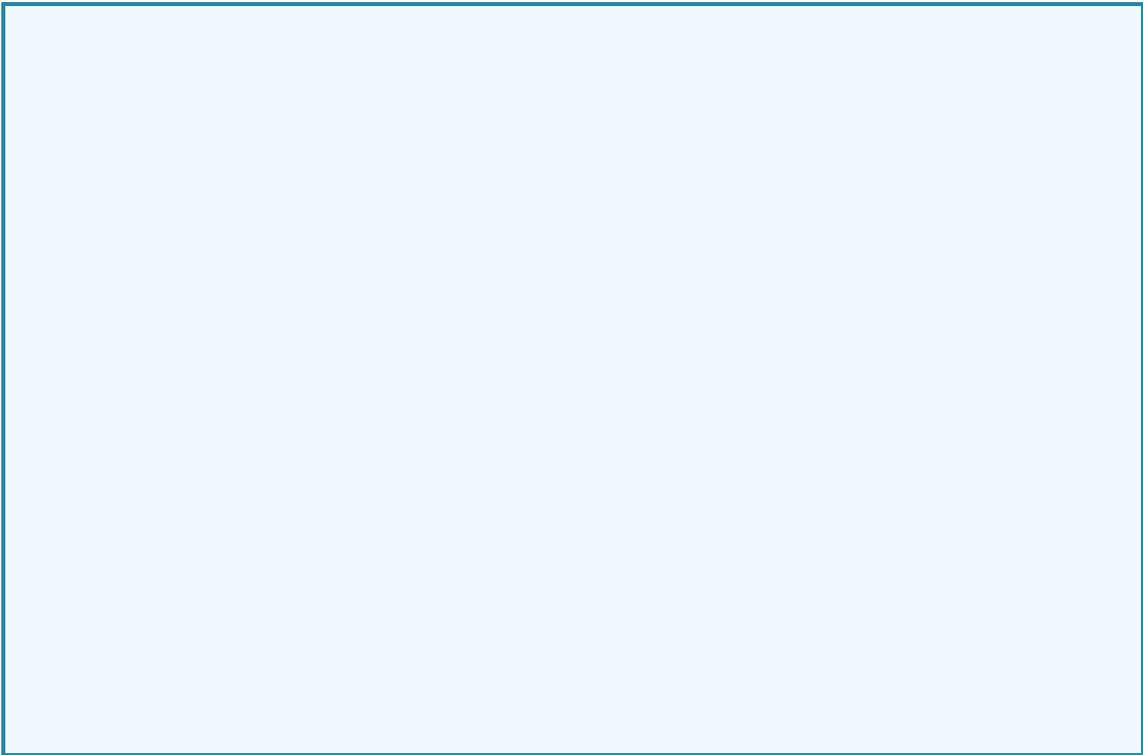
1. Avec une hauteur de 2



2. Avec une hauteur de 4



3. Avec une hauteur de 6



Exercice 3 : Quiz



Rappel

Un arbre binaire complet est un arbre binaire tel que chaque niveau de l'arbre est complètement rempli. Un arbre binaire complet de hauteur h contient donc $2^h - 1$ nœuds, et son nombre de feuilles est $F_h = 2^{h-1}$.

1. Quelle est, approximativement, la hauteur d'un arbre binaire complet à n nœuds ?
 - a) $O(\log n)$
 - b) $O(n/2)$
 - c) $O(2^n)$.
 - d) Ça dépend.
 - e) Aucune proposition ci-dessus n'est correcte.

2. Le parcours en profondeur d'un arbre binaire correspond à un fonctionnement de :

- a) file (First In First Out) ;
- b) pile (First In Last Out) ;
- c) liste chaînée.
- d) Ça dépend.
- e) Aucune proposition ci-dessus n'est correcte.

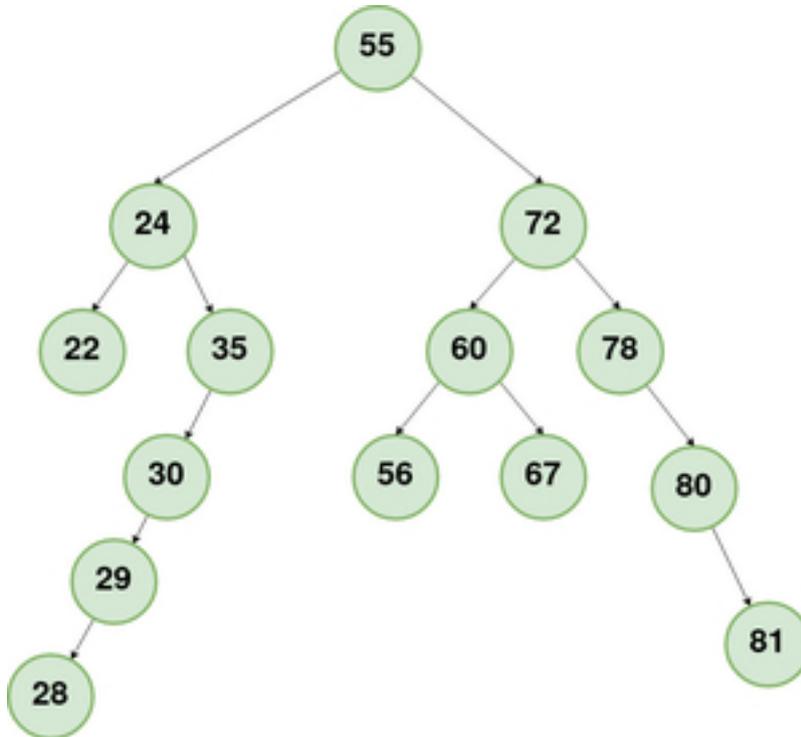
3. Une procédure Devine

```
1 PROCEDURE devine(a : arbre)
2 VARIABLE
3   Tmp : Arbre
4 DEBUT
5   SI Hauteur(a.filsG) > Hauteur(a.filsD) FAIRE
6     Tmp <- a.filsG
7     a.filsG <- a.filsD
8     a.filsD <- Tmp
9   FIN SI
10  devine(a.filsG)
11  devine(a.filsD)
12 FIN
```

Que fait la procédure **devine** ci-dessus ? On suppose qu'on dispose de la fonction *Hauteur()* qui renvoie la hauteur de l'arbre s'il est non vide.

- a) Elle échange le fils gauche et le fils droit de la racine si la hauteur du fils gauche est supérieure à la hauteur du fils droit, et les autres noeuds restent à leur place.
- b) Elle échange le fils gauche et le fils droit de la racine si la hauteur du fils gauche est supérieure à la hauteur du fils droit, et du coup les autres noeuds sont aussi échangés.
- c) Elle échange le fils gauche et le fils droit de tous les noeuds dont la hauteur du fils gauche est supérieure à la hauteur du fils droit, et les autres noeuds restent à leur place.
- d) Elle échange le fils gauche et le fils droit de tous les noeuds dont la hauteur du fils gauche est supérieure à la hauteur du fils droit, et du coup les autres noeuds sont aussi échangés.
- e) Aucune proposition ci-dessus n'est correcte.

4. Nous considérons l'arbre ci dessous :



Vérifier que l'arbre donné en exemple est bien un ABR en parcourant ses clés dans l'ordre infixe.

Exercice 4 : Des opérations

1. Donnez la structure permettant de créer un arbre binaire de recherche contenant des entiers.



2. Donnez l'algorithme permettant de rechercher un élément appartenant à un arbre binaire de recherche.



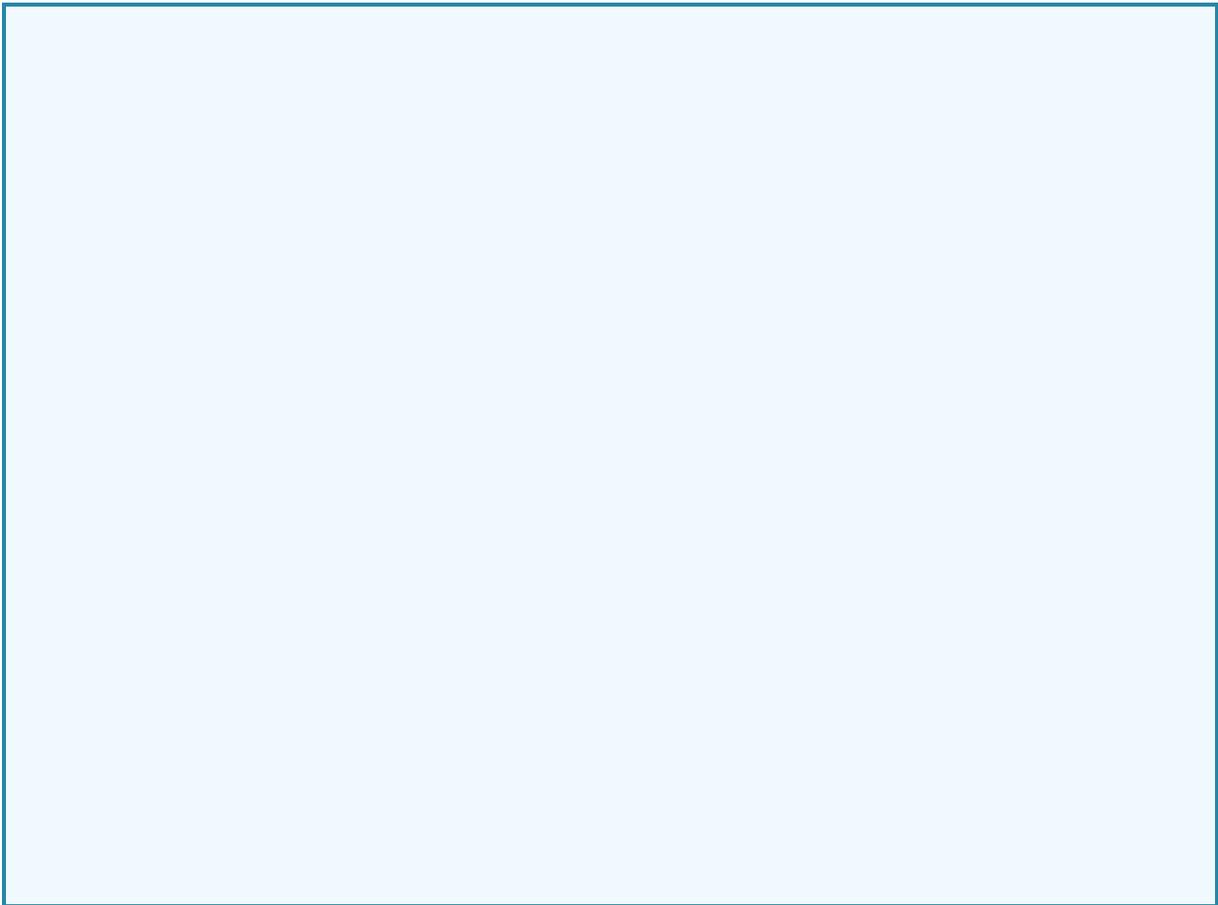
3. Donnez l'algorithme permettant de vérifier si un arbre binaire est un arbre binaire de recherche.



4. Donnez l'algorithme permettant de créer un arbre binaire de recherche de manière aléatoire. La fonction prendra en paramètre le nombre de noeud

```
1 FONCTION alea (nbDeNoeuds : entier, a : arbre) : arbre
```

On considère que l'on dispose de la fonction insertionArbre(valeur:entier,a:arbre):arbre



5. Donnez un algorithme permettant de réaliser l'intersection de deux arbres.

