

Arbres binaires de recherche

Dans ce TD, on va s'intéresser à la manipulation d'arbres binaires dont les étiquettes ont été ordonnées de sorte que la recherche d'une étiquette particulière soit facilitée : les arbres binaires de recherche.

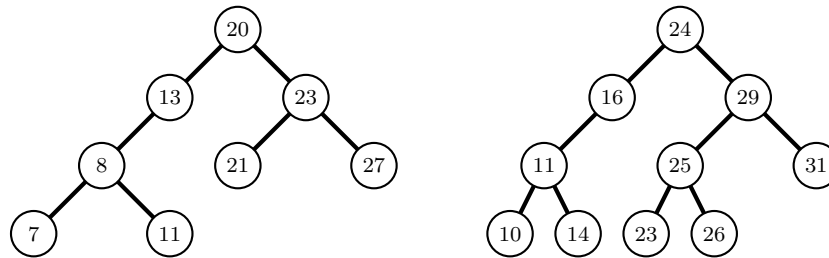


FIGURE 1 – Arbres binaires de recherche ou pas ?

1. Parcours

- Les arbres de la figure ci-dessus sont-ils des arbres binaires de recherche ?
- Donner la liste des valeurs correspondant aux parcours en profondeur en affichage préfixe, suffixe et infixe des deux arbres.
- Écrire une fonction `Arbre Recherche(Arbre a, int n)` qui recherche un élément `n` dans l'arbre `a`. Cette fonction renverra l'adresse du nœud contenant l'élément `n` s'il est présent, et renverra `NULL` sinon.

2. Ajout dans un arbre binaire de recherche — Dans cet exercice, on ne maintiendra et ne manipulera que des arbres binaires de recherche sans doublon.

- Construire l'arbre binaire de recherche obtenu par ajout successif des valeurs :

42, 18, 16, 12, 23, 45, 14, 13, 32, 62, 11.

Donner plusieurs ordres d'ajout pour lesquels l'arbre aura la plus grande hauteur possible, et un ordre d'ajout pour lequel l'arbre aura la plus petite hauteur possible.

- Écrire une fonction `int ajout(Arbre *a, int n)` qui effectue l'ajout de l'élément `n` dans l'arbre binaire de recherche `*a`. La fonction renverra `1` si l'ajout est réussi ou bien si `n` était déjà dans l'arbre, et renverra `0` en cas d'échec.

3. Vérification — Proposer différentes méthodes pour vérifier si un arbre binaire est un arbre binaire de recherche. Écrire les fonctions correspondantes.

4. Suppression dans un arbre binaire de recherche

- Supprimer successivement, dans l'arbre construit à la question précédente, les valeurs :

62, 16, 18.

Quel arbre obtient-on après chaque suppression ?

- Écrire une fonction `Arbre extraitMin(Arbre *a)` qui, étant donné un arbre binaire de recherche `*a`, effectue l'extraction du nœud contenant la plus petite étiquette de `*a`. Cette fonction devra modifier l'arbre `*a` et renvoyer l'adresse du nœud extrait, ou bien `NULL` en cas d'échec.
Que faut-il changer pour extraire le nœud contenant la plus grande étiquette ?
- Écrire une fonction `Arbre extrait(Arbre *a, int n)` qui extrait de l'arbre `*a` le nœud d'étiquette `n`. Cette fonction devra modifier l'arbre `*a` et renvoyer l'adresse du nœud extrait, ou bien `NULL` en cas d'échec.