

## TD n°2

### Arbre Binaire de Recherche

**Exercice 1** On suppose que les entiers compris entre 1 et 1000 sont disposés dans un arbre binaire de recherche, et on souhaite retrouver le nombre 363. Parmi les séquences suivantes, lesquelles ne pourraient pas être la séquence de nœuds parcourus ?

- a) 2, 252, 401, 398, 330, 344, 397, 363 ;
- b) 924, 220, 911, 244, 898, 258, 362, 363 ;
- c) 925, 202, 911, 240, 912, 245, 363 ;
- d) 2, 399, 387, 219, 266, 382, 381, 278, 363 ;
- e) 935, 278, 347, 621, 299, 392, 358, 363.

Est-il nécessaire de faire des schémas ? Écrire sous une forme minimale la propriété à vérifier.

**Exercice 2** Insérer successivement les entiers 7, 2, 9, 0, 5, 6, 8 et 1 dans un arbre binaire de recherche initialement vide. Que devient cet arbre après suppression de 2 puis de 7 ?

**Exercice 3** (*Recherche du successeur*) Soit  $A$  un arbre binaire de recherche dont les clés sont distinctes.

1. Montrer que l'élément minimum se trouve à un nœud sans fils gauche.
2. Montrer rigoureusement que si un nœud a un fils droit, son successeur est le minimum de son sous arbre droit. En déduire que si un nœud a un fils droit, alors son successeur n'a pas de fils gauche.
3. Montrer rigoureusement que si un nœud n'a pas de fils droit, son successeur, s'il existe, est le premier de ses ancêtres dont le fils gauche est aussi l'un de ses ancêtres.
4. Écrire un algorithme de recherche du successeur d'un nœud fourni en paramètre (qui retourne le sous arbre dont la racine est le successeur).
5. Quelle est la complexité en temps de cet algorithme ?

**Exercice 4** Soit  $A$  un arbre binaire de recherche dont les clés sont distinctes, soient  $x$  un nœud feuille et  $y$  son parent. Montrer que  $\text{clé}(x)$  est soit la plus petite clé de  $A$  supérieure à  $\text{clé}(y)$ , soit la plus grande clé de  $A$  inférieure à  $\text{clé}(y)$ .

**Exercice 5** On peut trier un ensemble donné de  $n$  nombres en commençant par construire par insertions successives un arbre binaire de recherche contenant ces nombres puis en effectuant un parcours infixe de l'arbre. Quels sont les temps d'exécution de cet algorithme dans le pire et dans le meilleur des cas ?