ARBRES BINAIRES DE RECHERCHE

À la fin de ce chapitre, je sais :

- ☞ parcourir un arbre de recherche pour trouver un élément
- ☞ insérer et supprimer un élément dans un arbre de recherche
- ☞ donner les complexités associées aux opérations sur un arbre binaire de recherche

A Définition et implémentation

■ **Définition 1 — Arbre binaire de recherche.** Soit un ensemble d'étiquettes $\mathcal{E}$ muni d'un ordre total. Un arbre binaire de recherche est un arbre binaire étiqueté par $\mathcal{E}$ dont les nœuds $N(e,g,d)$ vérifient la propriété suivante :

- l'élément e est plus grand que toutes les étiquettes du sous-arbre gauche g,
- l'élément e est plus petit que toutes les étiquettes du sous-arbre droit d.

■ **Exemple 1 — Arbres binaires de recherche.** La figure 1 représentent deux arbres binaires de recherche : l'un est étiqueté avec des entiers et est équilibré, l'autre est étiqueté avec des chaînes de caractères et n'est pas équilibré.

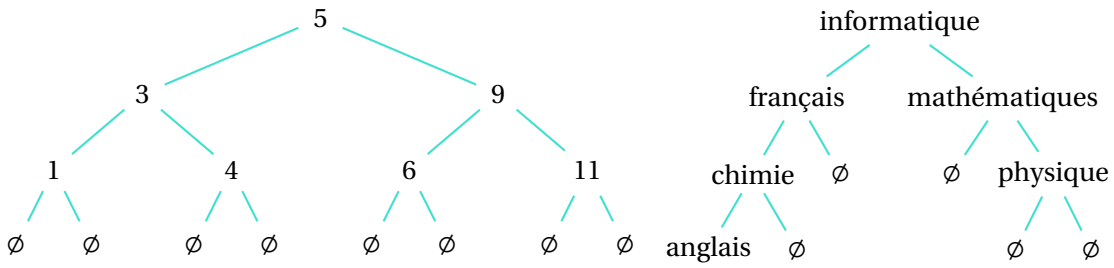


FIGURE 1 – Arbres binaires de recherche équilibré (à gauche) et non équilibré (à droite)

L'implémentation en OCaml s'appuie sur l'arbre binaire défini au chapitre précédent :

```
type 'a bst = Empty | Node of 'a * 'a bst * 'a bst
```

C'est un type **polymorphe**, c'est-à-dire que les étiquettes sont de type générique `'a` mais possèdent un ordre total, c'est-à-dire on doit pouvoir comparer n'importe quel élément de l'ensemble avec un autre.

B Opérations sur les arbres binaires de recherche

Les principales opérations sur les arbres de recherche sont :

1. la recherche d'un élément, opération de complexité $O(h)$,
2. l'insertion d'un élément, opération de complexité $O(h)$,
3. la suppression d'un élément $O(h)$.

R Si l'arbre est un peigne, ces opérations deviennent de complexité $O(n)$, si n est le nombre de nœuds de l'arbre. C'est le pire des cas.

Le meilleur des cas se produit lorsque l'arbre est équilibré : ces opérations sont alors donc de complexité $O(\log n)$.

Toute la question est donc d'opérer sur un arbre binaire de recherche et, **simultanément**, de le maintenir dans un état équilibré pour obtenir des performances optimales.

R L'insertion ou la suppression dans un arbre binaire de recherche garantit que la structure d'arbre binaire est respectée à la fin de l'opération. L'équilibre de l'arbre n'est pas garanti.

Les figure 2 et 3 illustrent les opérations d'insertion et de suppression dans un arbre binaire.

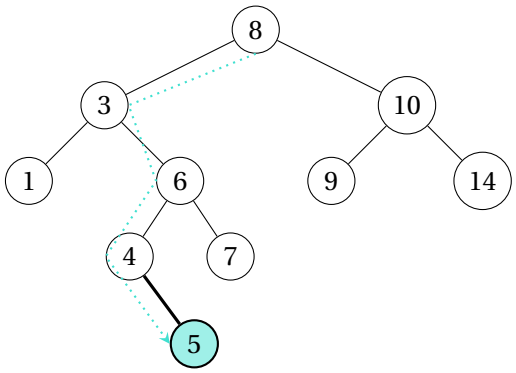
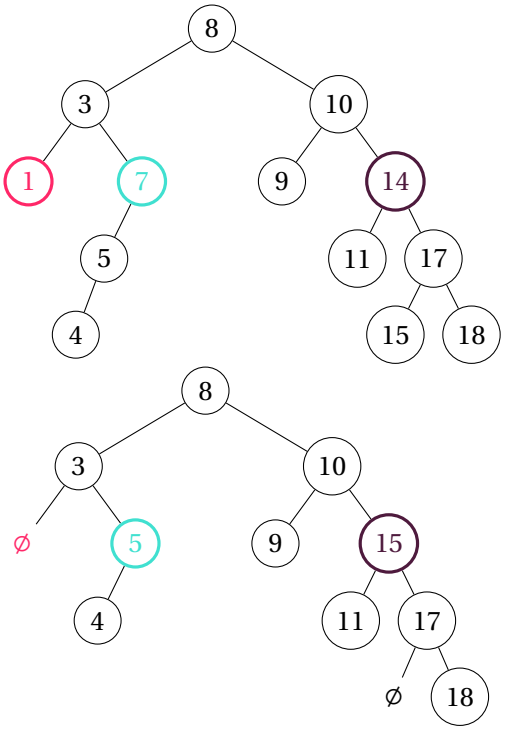


FIGURE 2 – Illustration de l'insertion d'un nœud d'étiquette 5 dans un arbre binaire de recherche



Trois cas sont possibles : le nœud qu'on retire

1. ne possède pas de fils,
2. possède un seul fils,
3. possède deux fils.

FIGURE 3 – Illustration de l'opération supprimer un nœud dans un arbre binaire de recherche

C Maintenir l'équilibre --→ HORS PROGRAMME

Le déséquilibre d'un arbre binaire équilibré est engendré par une opération d'insertion ou de suppression. Les techniques d'équilibrage s'appuient généralement sur les opérations d'insertion et de suppression normales suivies d'une manipulation de l'arbre pour lui redonner l'équilibre. Cette manipulation est souvent une rotation des sous-arbres.

■ **Exemple 2 — Arbres binaires de recherche automatiquement équilibrés.** On peut citer notamment :

1. Les arbres AVL,
2. Les arbres rouges et noirs.