

Dictionnaire arborescent

Objectifs

À la fin de cette séance, vous devriez être capable de :

- utiliser des arbres pour résoudre des problèmes ;
- parcourir des arbres de manière récursive ou itérative ;
- spécifier et prouver des algorithmes de parcours d'arbre.

Exercice 1.

On considère un type **Mot** qui définit une séquence de caractères. On souhaite construire un type **Dico**, permettant de gérer un ensemble de **Mot** (dictionnaire).

Les primitives que devra fournir ce type **Dico** sont :

- la recherche d'un **Mot** dans le dictionnaire
- l'insertion d'un nouveau mot dans le dictionnaire
- la suppression d'un nouveau mot dans le dictionnaire
- l'impression des mots du dictionnaire selon l'ordre lexicographique

Pour minimiser la mémoire nécessaire, on souhaite que l'implémentation de ce type **Dico** repose sur une structure arborescente permettant de «partager» les *préfixes* communs à plusieurs **Mot**.

- Q 1.* Dessinez un tel dictionnaire sous la forme d'un arbre (n-aire) contenant les mots suivants :
{arbitre, arbre, art, article, articuler, astre, bois, bon, botte, chat, zero}
- Q 2.* Pour faciliter l'implémentation, une approche possible consiste à transformer cet arbre n-aire en un arbre binaire. Proposez une telle transformation, et dessinez le dictionnaire obtenu sur l'exemple précédent.
- Q 3.* Proposez une structure de données basée sur du chaînage par pointeurs pour implémenter ce dictionnaire (sous la forme d'un arbre binaire).
- Q 4.* Spécifier et écrire une fonction **récursive** de recherche d'un mot dans le dictionnaire.
- Q 5.* Même question pour une fonction **itérative**
- Q 6.* Spécifier et écrire une fonction **récursive** d'insertion d'un mot dans le dictionnaire.
- Q 7.* Spécifier et écrire une fonction **récursive** de suppression d'un mot dans le dictionnaire.
- Q 8.* Spécifier et écrire une fonction d'impression du dictionnaire dans l'ordre lexicographique (récursive ou itérative, au choix).