

9.3 Les arbres de recherche

9.3.1 L'appartenance

Au chapitre 6, nous avons vu qu'un ensemble fini peut se représenter par une liste et qu'avec cette représentation, tester si un élément appartient à l'ensemble ou non demande en moyenne un temps linéaire en le nombre d'éléments de l'ensemble.

Cela revient à tester si un mot appartient à un dictionnaire en le comparant successivement, pour la relation d'égalité, à tous les mots du dictionnaire, depuis le premier jusqu'au dernier. On connaît une méthode plus efficace qui consiste à ouvrir le dictionnaire au milieu, à comparer, pour l'ordre alphabétique, le mot recherché au mot médian et à appliquer récursivement la même méthode au demi-dictionnaire obtenu.

Cette méthode de *recherche dichotomique* demande un temps logarithmique en la taille du dictionnaire. Cependant, elle repose sur le fait que les mots sont dans un certain ordre dans le dictionnaire et sur le fait que l'on peut avoir accès au mot médian en un temps indépendant de la taille du dictionnaire, ce qui est le cas si le dictionnaire est un codex, mais pas si c'est un volumen, comme un rouleau de papyrus.

Représenter un ensemble par une liste ne permet pas une telle recherche dichotomique, car même si l'on ordonne les éléments de la liste, accéder à l'élément médian d'une liste demande un temps linéaire, et non constant, en la taille de la liste.

En revanche, une telle recherche dichotomique est possible si l'on remplace la liste par un tableau. Mais cette solution a deux inconvénients. D'une part, la taille du tableau doit être choisie une fois pour toutes au moment de son allocation, et il n'est plus possible d'en changer si l'on veut faire évoluer la taille de l'ensemble. D'autre part, l'insertion d'un nouvel élément dans un tableau ordonné demande un temps linéaire en la taille du tableau, car cela nécessite de décaler tous les éléments supérieurs à l'élément inséré.

La solution pour pouvoir rechercher, insérer et supprimer un élément en temps logarithmique est d'utiliser un arbre de recherche. Un *arbre de recherche* est un arbre binaire dont chaque nœud contient un élément et tel que les éléments qui apparaissent dans le sous-arbre gauche d'un nœud sont inférieurs à l'élément contenu dans ce nœud et, de même, les éléments qui apparaissent dans le sous-arbre droit d'un nœud sont supérieurs à l'élément contenu dans ce nœud. Un même ensemble peut se représenter par un arbre de recherche de plusieurs manières. Par exemple, l'ensemble {1, 5, 9, 12, 19, 24} peut se représenter par les arbres de recherche