

Arbre AVL

Dans ce TP il s'agit d'implémenter un dictionnaire à l'aide d'arbres AVL (du noms de leurs créateurs). L'interface et les fonctions à implémenter sont donc les mêmes que pour le TP précédant (les fichiers `avl.h` et `map.h` sont identiques). Les arbres AVL sont des arbres binaires de recherche particuliers : ils sont *équilibrés*. Plus précisément, pour chaque nœud la différence de *hauteur* entre ses fils est au plus un. Maintenir un arbre AVL après une opération d'ajout ou de suppression consiste à effectuer un rééquilibrage de l'arbre à l'aide de rotation.

Il existe plusieurs définitions de la hauteur d'un arbre, mais en général pour les AVLs on utilise comme définition le nombre maximum de nœuds d'un chemin allant de la racine à l'une de ses feuilles. En particulier un arbre vide a pour hauteur 0 et un arbre à un seul nœud (ou feuille) a pour hauteur 1.

1 Implémentation récursive d'un AVL

Donner une première implémentation récursive AVL.

2 Implémentation itérative d'un AVL

En utilisant la structure de pile implémentée au TP1, montrer comment simuler les appels récursifs dans une boucle `while` et donner ainsi une implémentation itérative d'un AVL.