



et une *rotation gauche* est l'inverse d'une rotation droite. De telles rotations transforment des arbres de recherche en des arbres de recherche.

Exercice 9.5

On considère un arbre tel que les deux sous-arbres de la racine soient AVL mais tel que la différence de hauteur de ces deux sous-arbres soit égale à 2.

Montrer que l'on peut transformer cet arbre en un arbre AVL en appliquant des rotations à la racine et à ses enfants.

On distinguera plusieurs cas selon la hauteur des arbres dont les racines sont les enfants et les petits-enfants de la racine de l'arbre à équilibrer.

Exercice 9.6

Écrire en Java des fonctions d'insertion et de suppression dans les arbres AVL. Comment éviter de devoir calculer la hauteur d'un arbre — ce qui demande un temps linéaire et non logarithmique en sa taille ?

9.3.3 Les dictionnaires

Au chapitre 6 nous avons vu qu'une liste d'associations est une représentation d'une fonction de domaine fini, typiquement un dictionnaire, sous la forme d'une liste de couples telle que pour tout k , il existe au plus un élément v tel que le couple (k, v) appartienne à la liste.

La recherche de la valeur associée à une clé dans une liste d'associations est linéaire en le nombre de couples. En remplaçant la liste par un arbre de recherche ordonné par l'ordre sur les clés, on peut rendre la recherche, l'insertion et la suppression logarithmiques.