

la case i du tableau. Les enfants du nœud de la case i — nœud de numéro $i + 1$ — sont dans les cases $2i + 1$ et $2i + 2$ et le parent du nœud de la case i est dans la case $(i - 1) / 2$.

Si on suppose que les cases $0 \dots i - 1$ d'un tableau représentent un tas, l'insertion de l'élément situé dans la case i du tableau s'écrit ainsi

```
static void insert (final int [] t, final int i) {
    int j = i;
    while (j >= 1 && t[(j-1)/2] <= t[j]) {
        int z = t[j]; t[j] = t[(j-1)/2]; t[(j-1)/2] = z; j = (j-1)/2; } }
```

Et si on suppose que les cases $0 \dots i$ d'un tableau représentent un tas, la suppression de l'élément situé dans la case 0 du tableau s'écrit ainsi

```
static void suppress (final int [] t, final int i) {
    t[0] = t[i];
    int j = 0;
    while ((2*j+1 < i && t[j] <= t[2*j+1])
        || (2*j+2 < i && t[j] <= t[2*j+2])) {
        int m = 2*j+1;
        if (m + 1 < i && t[m + 1] >= t[m]) m = m + 1;
        int z = t[j]; t[j] = t[m]; t[m] = z;
        j = m; } }
```

Exercice 9.9

On peut également représenter les files de priorité par des arbres AVL. Programmer la recherche de l'élément de priorité maximale, l'insertion et la suppression de l'élément de priorité maximale dans un arbre AVL, en temps logarithmique.

Exercice 9.10 (La compression d'images)

La manière la plus simple de représenter une image est sous forme d'un tableau de nombres, chaque nombre étant la couleur d'un pixel. Ainsi, une image en noir et blanc, de 256 pixels par 256 pixels peut se représenter par un tableau de 65536 nombres qui valent chacun 0 ou 1.

Cependant, avec une image comme celle-ci