Algorithmique
TD 8. Semaine du 3/12/2001

Fonctions de Hachage

Exercice 1

Etant donné un entier m, on considère la fonction de hachage h définie sur les entiers par:

$$h(x) = x \bmod m$$

1. Soit k un entier non nul. Quelle est la probabilité pour que deux entiers x, $y \in [0 \ldots k \times m[$ verifient $h(x) = h(y)$?
2. Soit $p \geq 2$. On suppose $m = 2^p$. Quelle est la condition sur l'écriture en binaire de deux entiers x et y pour que l'on ait $h(x) = h(y)$?
3. On suppose à présent que $m = 2^p - 1$. En admettant les résultats démontrés à l'exercice 2,
   - montrer que si l'on change la valeur d'un seul bit de l'écriture en binaire de x, on obtient une valeur de h différente.
   - montrer que l'échange d'un 0 et d'un 1 adjacents change également cette valeur.

Exercice 2

Soit $p \geq 2$. Montrer que

$$x \bmod (2^p - 1) \neq (x + 2^k) \bmod (2^p - 1)$$

Faire le même raisonnement pour

$$x \bmod (2^p - 1) \neq (x - 2^k) \bmod (2^p - 1)$$

Exercice 3

On considère la méthode suivante, permettant de stocker un entier `n` dans un tableau `tab`. Les éléments de `tab` seront supposées initialisés à -1 au moment du premier appel. La méthode `static int hash(int m, int n)` est à valeurs dans $\{0, \ldots, \mathtt{m}-1\}$.

```
static void ajouter(int[] tab, int n){
  int m = tab.length, i = hash(m,n),
      j = 0;

  while (tab[i] > -1 &&
         tab[i] != n &&
         j < m) {
     i = i+1 % m;
     j++;
  }

  if (tab[i] == - 1) tab[i] = n;
}
```

Analyser le fonctionnement de cette méthode. En déduire une méthode `static boolean detecter(int[] tab, int n)` renvoyant `true` si et seulement si `n` a déjà été ajouté au tableau.

Arbres Binaires en Java

Implémenter en Java la classe `Arbre` des arbres binaires. On utilisera la technique de l'encapsulage, de manière à ce que l'arbre vide soit aussi un objet. La classe `Arbre` contiendra au minimum les méthodes suivantes :

- `boolean estVide()` renvoyant `true` si et seulement si `this` est vide.
- `Arbre gauche()` et `Arbre droit()` renvoyant les sous-arbres gauche et droit de `this`.
- `int hauteur()` renvoyant la hauteur de `this`.