Travaux Dirigés d'algorithmique n°9

Cours d'Informatique de Deuxième Année

—L2.1—

Manipulation de bits et tableaux de listes

Dans ce TD, nous apprenons à manipuler des bits en C, en utilisant les opérateurs logiques et les masques. Nous conclurons par un exercice visant à stocker des entiers de façon efficace dans des tableaux de listes chaînées.

► Exercice 1. Manipulation de bits

Nous souhaitons écrire des fonctions de base pour manipuler des bits. En C, on use des opérateurs logiques classiques (&, |, ^, ~, «, ») et des masques pour ce genre d'opérations.

ATTENTION : nous utilisons des entiers, et nous indiquerons le bit de poids fort par 31, et celui de poids faible par 0.

Vous écrirez les fonctions suivantes :

- `strToBin` : on passe à cette fonction une chaîne de caractères de longueur maximum 32. Elle vérifie que cette chaîne n'est composée que de 0 et de 1, puis renvoie l'entier dont la représentation binaire correspond à la chaîne saisie.
- `afficheBin` : elle reçoit un entier et l'affiche en représentation binaire. Écrivez une version itérative et une version récursive de cette fonction.
- `testeBit` : permet de savoir la valeur d'un des bits d'un entier. Elle renvoie 0 ou 1.
- `allumeBit` : positionne un bit donné d'un entier à 1.
- `eteintBit` : positionne un bit donné d'un entier à 0.
- `estPalindrome` : renvoie VRAI ou FAUX selon que l'entier passé en paramètre a une représentation binaire palindromique ou non.

► Exercice 2. Tableaux de listes chaînées

Le but de cet exercice est de stocker des entiers dans des listes chaînées, en manipulant des cellules simples contenant un `int`.

1. Définissez un type `Cellule` adéquat.
2. Écrivez rapidement les fonctions `ajouteEntier` et `chercheEntier`. Calculez leur complexité.

3. *Par quel moyen pourriez-vous accélérer la recherche sans augmenter la complexité de l'ajout ?*
4. *Dans le cas du stockage de nombres binaires, quelle fonction de répartition choisir ?*
5. *La fonction de répartition est-elle importante ? Sur quel(s) critère(s) doit-t'elle être choisie ? Peut-on imaginer une optimisation similaire sur le stockage de chaînes de caractères ?*