

Algorithmique

Travaux dirigés, 11 octobre 2005

Louis Granboulan

1 Systèmes de numération redondants

La représentation classique en base B d'un entier n sous la forme $n = \sum_{i \geq 0} n_i B^i$ avec $0 \leq n_i < B$ n'est pas la seule intéressante. On étudie ici la représentation $n = \sum_{i=0..k-1} n_i B^i$ avec $-c_{\min} \leq n_i \leq c_{\max}$ avec c_{\min} et $c_{\max} < B$.

1. Donner une condition nécessaire et suffisante sur les entiers naturel B , c_{\min} et c_{\max} pour que tout entier naturel admette une représentation. Donner un exemple de système de numération redondant, i.e. tel que certains entiers admettent plusieurs représentations.
2. Supposant que $c_{\min} = c_{\max} = a$, que fait l'algorithme suivant qui prend $x = (x_{k-1} \dots x_0)$ et $y = (y_{k-1} \dots y_0)$ pour fournir les $k + 1$ valeurs z_0, \dots, z_k ?
 1. Pour $i = 0..k - 1$ calculer $t_{i+1} = \begin{cases} 1 & \text{si } x_i + y_i \geq a \\ -1 & \text{si } x_i + y_i \leq -a \\ 0 & \text{si } x_i + y_i \in]-a, a[\end{cases}$
 2. Pour $i = 0..k$ calculer $z_i = w_i + t_i$ avec la convention $t_0 = w_k = 0$.
3. Quel est l'intérêt d'un tel algorithme ?

2 Exponentiation

1. Algorithmes itératifs

Dans un monoïde multiplicatif G , on suppose que les opérations `mul` et `sqr` de multiplication et d'élevation au carré sont efficaces. On cherche à calculer la puissance n -ième.

- **Square-and-multiply.** Décrire un algorithme qui effectue ce calcul en $\mathcal{O}(\log n)$ opérations. Compter précisément le nombre de carrés et de multiplications.
- **Précalculs.** Imaginer une variation de cet algorithme qui, avec un petit précalcul, diminue le nombre moyen de multiplications.

2. Cas des groupes

Dans le cas où G est un groupe, on peut espérer faire mieux si le calcul de l'inverse est efficace.

- **Représentation signée.** Trouver un algorithme qui donne une représentation de longueur minimale de l'entier n comme somme signée de puissances de 2.
- **Application.** Utiliser ceci pour le calcul de la puissance n -ième. Quelle est la complexité, exprimée en carrés, multiplications et divisions.

3. Chaînes d'additions

Une chaîne d'additions pour n est une suite a_0, \dots, a_r telle que $a_0 = 1$, $a_r = n$ et tout élément de la liste est la somme de deux éléments précédents ($a_i = a_j + a_k$ avec $j, k < i$). On appelle $l(n)$ la longueur de la plus petite chaîne d'additions pour n .

- (a) Exprimer quelles propriétés de $l(n)$ on peut déduire des méthodes précédentes.
- (b) On appelle doublage un élément d'une chaîne d'additions de la forme $a_i = a_{i-1} + a_{i-1}$. Soit d le nombre de doublages d'une chaîne d'additions de longueur r , pour n . Montrer que $n \leq 2^{d-1} F_{r-d+3}$ où (F_k) est la suite de Fibonacci.
- (c) Trouver $l(n)$ si $n = 2^A$, $n = 2^A + 2^B$, $n = 2^A + 2^B + 2^C$.