Preuve et Analyse des Algorithmes

14:00–16:00
aucun document autorisé

4 janvier 2016

1 Stabilité d'un algorithme de tri

Un algorithme de tri qui compare et déplace les éléments d'un tableau est dit stable s'il ne change pas l'ordre relatif de deux éléments égaux pour la clé de tri considérée. Autrement dit, si $t[i] = t[j]$ pour deux indices $i < j$ à l'entrée de l'algorithme de tri, alors la position finale de l'élément $t[i]$ est inférieure à celle de $t[j]$.

1. Donner un exemple d'algorithme de tri stable.
2. Compléter la ligne 10 de la coupure de Lomuto.
3. Etudier la stabilité du tri basé sur Lomuto.

2 Résidu primitif

Soit m un nombre premier. On appelle résidu modulo m tout entier naturel inférieur à m. Pour tout entier naturel z, il existe un et un seul résidu, que nous noterons $[z]_m$, tel que

$$\exists q \in \mathbf{N}, \quad z = qm + [z]_m \tag{1}$$

On rappelle que la multiplication modulaire $(x, y) \mapsto x \otimes y := [xy]_m$ définit un opérateur binaire associatif sur les résidus modulo m. Un résidu g est dit primitif modulo m si pour tout résidu non nul y modulo m, il existe un entier k tel que $y = [g^k]_m$, on dit alors que k est le logarithme à base g de y. Un résultat classique de la théorie des nombres, que nous admettrons, affirme que tout nombre premier m admet au moins un élément primitif.

1. Quelle notion mathématique permet d'affirmer l'existence et l'unicité du résidu décrit dans la relation (1).
2. Quels sont les résidus primitifs modulo 7 ?

```
1  procedure split(x, left, right, i)
2  T := x[left];
3  i := left;
4  for j := left + 1 to right do
5    if x[j] < T  then
6   i := i + 1
7   swap(x[i], x[j])
8    fi
9  done
10 // ???
11 end. {split}
```

Lomuto

```
1 char val( ullong z )
2 {
3 int r = 0;
4 while ( ... ) {
5         ...
6         ...
7 }
8 return r;
9 }
```

3. Montrer que si x est un résidu quelconque alors il existe deux entiers $i < j$ tel que $[x^i]_m = [x^j]_m$, sans que m ne soit nécessairement premier.
4. Montrer que si x est un résidu non nul (m premier) alors il existe un entier $k > 0$ tel que $[x^k]_m = 1$.
5. Montrer que g est primitif si et seulement si

$$\forall k, \qquad 0 < k < m - 1 \quad \Longrightarrow \quad [g^k]_m \neq 1.$$

6. Ecrire une fonction `primitif(int m)` qui renvoie le plus petit résidu primitif modulo m.
7. Ecrire une fonction `int dl (int y, int g, int m)` qui détermine le logarithme de y en base g sachant que g est primitif modulo m.
8. Préciser le temps de calcul moyen.

3 Logarithme des mots de poids 1

Un mot binaire de poids 1, s'écrit 2^v pour un certain v. Dans cet exercice, on s'intéresse aux fonctions écrites en Langage C qui déterminent l'entier v pour des entiers de 64 bits mais de poids 1.

1. Donner un exemple de contexte d'utilisation.
2. Compléter la fonction `char val(ullong z)` en privilégiant l'utilisation des opérateurs bit-à-bit du Langage C.
3. Le type de retour de la fonction `val` est-il correct ?
4. Préciser les temps de calcul.
5. Ecrire une fonction `int log(ullong z)` plus efficace.
6. Préciser le temps de calcul.
7. Décrire une solution basée sur le fait que 2 est primitif modulo 67.