Université de Bordeaux — **N1MA8W02 – Master**
Mathématiques — Année 2016–2017

Devoir Surveillé, 8 mars 2017 : Corrigé

Durée 1h30.

À la fin de l'épreuve, votre fichier "`votre_nom.sage`" ou "`votre_nom.sws`" est à envoyer à l'adresse : `arnaud.jehanne@u-bordeaux.fr`

Exercice 1 – [Suite de Fibonacci]
Soit $(F_n)_{n\in\mathbb{N}}$ la suite de fibonacci definie par $F_0 = 0$, $F_1 = 1$ et $F_n = F_{n-1} + F_{n-2}$ pour tout $n \geqslant 2$.

1) *Montrer que pour tout $(n, k) \in \mathbb{N}^2$, $F_{n+k+1} = F_nF_k + F_{n+1}F_{k+1}$.*

Pour n fixé, on raisonne par récurrence sur k. Vérifions d'abord l'égalité pour $k = 0$ et $k = 1$.
$k = 0$: $F_{n+1} = F_nF_0 + F_{n+1}F_1$ puisque $F_0 = 0$ et $F_1 = 1$.
$k = 1$: $F_{n+2} = F_{n+1} + F_n = F_nF_1 + F_{n+1}F_2$ puisque $F_1 = 1$ et $F_1 = 1$.

Soit $k \geqslant 2$. On suppose que pour tout entier $l < k$, $F_{n+l+1} = F_nF_l + F_{n+1}F_{l+1}$. Alors $F_{n+k+1} = F_{n+k} + F_{n+k-1}$. L'hypothèse de récurrence donne l'égalité

$$F_{n+k+1} = F_nF_{k-1} + F_{n+1}F_k + F_nF_{k-2} + F_{n+1}F_{k-1}$$

(remarquons la donnée $k \geqslant 2$: elle assure que $k - 2 \geqslant 0$ et que l'hypothèse de récurrence s'applique bien). On en déduit

$$\begin{aligned} F_{n+k+1} &= F_n(F_{k-1} + F_{k-2}) + F_{n+1}(F_k + F_{k-1}) \\ &= F_nF_k + F_{n+1}F_{k+1} \end{aligned}$$

2) *Soit n un entier strictement positif. Montrer que $F_{2n+1} = F_n^2 + F_{n+1}^2$ et que $F_{2n} = 2F_nF_{n+1} - F_n^2$.*

En appliquant l'égalité précédente à $k = n$, on obtient directement $F_{2n+1} = F_n^2 + F_{n+1}^2$. Appliquons maintenant cette même égalité à $n = k - 1$. On obtient $F_{2n} = F_nF_{n-1} + F_{n+1}F_n = F_n(F_{n-1} + F_{n+1}$. Comme $F_{n-1} = F_{n+1} - F_n$, on arrive bien au résultat demandé.

3) *En utilisant ces deux égalités, écrire un algorithme* `Fibo` *qui calcule F_n en $O(\log n)$ opérations dans $\mathbb{Z}$.*

Soit $C(n)$ le nombre d'opérations dans $\mathbb{N}$ dans l'exécution de `Fibo(n)`. Alors

$$C(n) \leqslant C\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 7.$$

Remarquons que si $\overline{a_k \dots a_0}$ (où $k = \lfloor \log n \rfloor$) est l'écriture de n en binaire, celle de $\left\lfloor \frac{n}{2} \right\rfloor$ est $\overline{a_k \dots a_1}$. On en déduit que $C(n) \leqslant 7 \lfloor \log n \rfloor$.

Exercice 2 – [Restes chinois]
Notation. *Dans cet énoncé, si $m \in \mathbb{N} \setminus \{0,1\}$, tout entier noté $\mod m$ est pris dans l'intervalle $[[0, m-1]]$. En particulier, on note respectivement $z^{-1} \mod m$ et $yz^{-1} \mod m$ les entiers k et l de $[[0, m-1]]$ tels que $kz \equiv 1 \mod m$ et $l \equiv yk \mod m$.*
Rappel. [Théorème des restes chinois] *Soit $n \in \mathbb{N} \setminus \{0\}$. Soient $m_0, \ldots, m_{n-1}$ des entiers de $\mathbb{N} \setminus \{0\}$ deux à deux premiers entre eux, et soient $a_0, \ldots, a_{n-1}$ des entiers quelconques. Le système de congruences*

$$x \equiv a_i \mod m_i \quad (i \in [[0, n-1]]) \tag{1}$$

admet une solution

$$x = \sum_{i=0}^{n-1} (a_i M_i^{-1} \mod m_i) M_i \tag{2}$$

où

$$M = \prod_{i=0}^{n-1} m_i \quad \text{et} \quad M_i = \prod_{j \neq i} m_j \quad \text{pour tout } i \in [[0, n-1]]$$

De plus, cette solution est unique modulo M.

1) *Écrire sur votre fichier ou session sage une fonction qui étant donnés les entiers $m_0, \ldots, m_{n-1}, a_0, \ldots, a_{n-1}$, utilise l'égalité précédente pour donner la solution de (1) qui appartient à $[[0, M-1]]$.*
Commandes sage. *Pour calculer $y^{-1} \mod m$, on peut écrire* `mod(y,m)**(-1)`. *On obtient un élément de $\mathbb{Z}/m\mathbb{Z}$. Si* `s` *définit un élément de $\mathbb{Z}/m\mathbb{Z}$, alors la commande* `lift(s)` *(ou* `s.lift()`*) en donne un relèvement dans $\mathbb{Z}$.*

2) *Appliquer cette fonction aux données $n = 4$, $m_0 = 5$, $m_1 = 7$, $m_2 = 11$, $m_3 = 13$, $a_0 = 2$, $a_1 = 1$, $a_2 = 3$, $a_3 = 8$. Si votre fonction fait défaut, appliquer la fonction sage* `crt` *à ces mêmes données. Pour cette question, contentez-vous d'écrire le résultat sur votre copie.*

On trouve 2192.

Nous allons maintenant nous intéresser à un autre algorithme pour résoudre un tel système.

3) *On considère l'algorithme suivant.*

Algorithme 1. Algorithme de Garner

Entrées: les m_i, les a_i
Sorties: $x \in [[0, M-1]]$ solution de (1)

pour $i = 1$ à $n-1$ **faire**
  $C_i = 1$
  pour $j = 0$ à $i-1$ **faire**
    $u = m_j^{-1} \mod m_i$
    $C_i = uC_i \mod m_i$
$u = a_0 \mod m_0$
$x = u$
pour $i = 1$ à $n-1$ **faire**
  $u = (a_i - x)C_i \mod m_i$
  $x = x + u\prod_{j=0}^{i-1} m_j$
Retourner x

Exécuter à la main cet algorithme avec les données de la question 2), c'est-à-dire $n = 4$, $m_0 = 5$, $m_1 = 5$, $m_2 = 11$, $m_3 = 13$, $a_0 = 2$, $a_1 = 1$, $a_2 = 3$, $a_3 = 8$.

Laissé au lecteur.

4) *Pour* $i \in [[1, n-1]]$, *exprimer* C_i *en fonction des* m_j.

$C_i = \prod_{j=0}^{i-1} m_j^{-1} \mod m_i$.

5) *Soit pour tout* i *l'entier* $P_i = \prod_{j=0}^{i} m_j$. *On note* x_0 *la valeur atteinte par* x *au pas* 7 *et pour* $i \in [[1, n-1]]$, *on note* x_i *la valeur de* x *après le* i*-ème passage dans la boucle "pour" du pas* 8.

Montrer que pour tout i, *l'entier* x_i *appartient à* $[[0, P_i - 1]]$ *et vérifie les congruences*

$$x_i \equiv a_j \mod m_j \quad \textit{pour tout} \quad j \in [[0, i]].$$

En déduire que l'algorithme de Garner donne bien l'unique solution x *du système (1) qui appartient à l'intervalle* $[[0, M-1]]$.

On note $u_0 = a_0 \mod m_0$ et pour tout i, on note u_i la valeur de u_i obtenue après le pas i.

Il est clair que $x_0 \in [[0, m_0 - 1]]$ et que $x_0 \equiv a_0 \mod m_0$. Supposons par récurrence que pour $i \geqslant 1$ fixé, $x_{i-1} \in [[0, P_{i-1} - 1]]$ et que

$$x_{i-1} \equiv a_j \mod m_j \quad \text{pour tout} \quad j \in [[0, i-1]].$$

Alors

$$\begin{aligned} x_i &= x_{i-1} + u_{i-1}\prod_{j=0}^{i-1} m_j \\ &\leqslant (P_{i-1} - 1) + (m_i - 1)P_{i-1} \\ &\leqslant P_i - 1 \end{aligned}$$

De plus, pour tout $j \leqslant i-1$, $x_i \equiv x_{i-1} \mod m_j$, donc $x_i \equiv a_j \mod m_j$. Voyons la congruence modulo m_i.

$$\begin{aligned} x_i &\equiv x_{i-1} + u_i P_{i-1} \mod m_i \\ &\equiv x_{i-1} + (a_i - x_{i-1}) C_i P_{i-1} \mod m_i. \end{aligned}$$

Or $C_i P_{i-1} \equiv 1 \mod m_i$, donc $x_i \equiv a_i \mod m_i$.

6) *Quel est l'avantage de cet algorithme sur le calcul direct utilisant l'égalité (2) ?*

Dans le calcul direct, les calculs intermédiaires peuvent mener à des entiers supérieurs à M. Dans l'algorithme de Garner, on reste toujours dans l'intervalle $[[0, M-1]]$. De plus, à chaque pas i, les entiers qui interviennent dans les calcul sont majorés par P_i.

7) *Écrire l'algorithme de Garner sur votre fichier ou session sage.*