DM D'INFORMATIQUE (Option) n°2

À rendre le mercredi 13/06/18

⋆⋆⋆

Ce sujet traite du problème du monnayeur : comment rendre la monnaie en utilisant le plus petit nombre possible de pièces ?
On appelle **système** un m-uplet $c = (c_1, \dots, c_m)$ d'entiers vérifiant $c_1 > c_2 > \dots > c_m = 1$. Par exemple, dans le système Euro, les pièces forment le système $(200, 100, 50, 20, 10, 5, 2, 1)$.
Si $x \in \mathbb{N}$ est un montant à rendre, une représentation de x dans c est un m-uplet $p = (p_1, \dots, p_m)$ vérifiant $x = \sum_{i=1}^{m} p_i c_i = p \cdot c$. Dans cette représentation, p_i représente donc le nombre de pièces c_i qui seront rendues.

Nous souhaitons minimiser la quantité de pièces rendues, c'est-à-dire la quantité $\|p\| = \sum_{i=1}^{m} p_i = p \cdot \mathbb{1}$ où $\mathbb{1} = (1, \dots, 1)$.

Nous noterons $M_c(x)$ (ou $M(x)$ s'il n'y a pas d'ambiguïté) le plus petit nombre de pièces nécessaires pour représenter x, c'est-à-dire $M(x) = \min\{\|p\|, p \cdot c = x\}$.

I Système et poids minimal

Un système ou une représentation sera représentée par un tableau d'entier en Caml. Par exemple, `[|4;1;3|]` est une représentation de 30 dans le système `[|6;3;1|]`.

Question 1 Écrire une fonction `systeme : int array -> bool` qui teste si un tableau `c` correspond à la définition d'un système.

Question 2 Donner un exemple de système c et de valeur $x \in \mathbb{N}^*$ ayant plusieurs représentations dans ce système.

Question 3 Montrer que $\left\lceil \frac{x}{c_1} \right\rceil \leq M(x) \leq x$.

Question 4 Soit $x \in \mathbb{N}^*$. Soit k le plus petit indice i tel que $c_i \leq x$. Montrer que $M(x) = 1 + \min_{k \leq i \leq m} M(x - c_i)$.

Question 5 Écrire une fonction `poids_min : int -> int array -> int array` qui prend en argument un entier x et un système c et renvoie le tableau des $M_c(y)$ pour $y \in [\![0; x]\!]$.

Question 6 Déterminer la complexité temporelle de la fonction précédente.

Question 7 Comment modifier la fonction précédente pour renvoyer p tel que $\|p\| = M(x)$?

II Algorithme glouton et système canonique

L'algorithme glouton pour rendre une somme $x \in \mathbb{N}$ consiste à choisir le plus grand $c_i \leq x$, puis à rendre récursivement $x - c_i$ en plus de la pièce c_i. Par exemple, avec le système $c = (10, 5, 2, 1)$ et $x = 27$, l'algorithme glouton choisira $p = (2, 1, 1, 0)$. Nous noterons par la suite $\Gamma_c(x)$ (ou $\Gamma(x)$ s'il n'y a pas d'ambiguïté) la solution renvoyée par cet algorithme et $G_c(x)$ (ou $G(x)$) le nombre de pièces utilisées : $G(x) = \|\Gamma(x)\|$.

Question 8 Écrire une fonction **récursive** (ou utilisant une fonction auxiliaire récursive) `glouton : int -> int array -> int array` qui calcule une solution par l'algorithme glouton.

Un système c est dit **canonique** si l'algorithme glouton renvoie toujours une représentation minimale, c'est-à-dire si $\forall x \in \mathbb{N}, M(x) = G(x)$. Si c n'est pas canonique, on dit que x est un **contre-exemple** pour c si $M(x) < G(x)$.

Question 9 Montrer qu'un système $c = (c_1, c_2)$ est toujours canonique.

Question 10 Montrer qu'il existe un système $c = (c_1, c_2, c_3)$ non canonique, en justifiant par un contre-exemple.

Question 11 Soient q et n deux entiers ≥ 2. Montrer que $(q^n, q^{n-1}, \dots, q, 1)$ est canonique.

Question 12 Montrer que le système Euro est canonique.

Question 13 Avant la réforme de 1971 (introduisant un système décimal), le Royaume-Uni utilisait le système $(30, 24, 12, 6, 3, 1)$. Montrer que ce système n'est pas canonique.

Question 14 Soit c un système non canonique et x le plus petit contre-exemple. Montrer que $c_{m-2}+1 < x < c_1+c_2$.

Question 15 Soit $q \geq 3$. Montrer que $(q+1, q, 1)$ n'est pas canonique. Déterminer le plus petit contre-exemple pour ce système.

Question 16 Soit $q \geq 3$. Déterminer $f(q)$ tel que $(f(q), q, 1)$ ne soit pas canonique et admette $f(q)+2$ comme plus petit contre-exemple.

Question 17 Déduire des questions précédentes une fonction `canonique : int array -> bool` qui teste si un système c est canonique ou non.

⋆ ⋆ ⋆