DM D'INFORMATIQUE (Option) n°2

Corrigé

⋆⋆⋆

I Système et poids minimal

Question 1 On vérifie que chaque élément est plus grand que le suivant et que le dernier vaut 1.

```
let systeme c =
   let b = ref true in
   for i = 1 to Array.length c - 1 do
      if c.(i-1) <= c.(i) then b := false
   done;
   !b && c.(Array.length c - 1) = 1;;
```

Question 2 Si on pose $c = (5, 2, 1)$ et $x = 5$, alors deux représentations possibles pour x sont $(1, 0, 0)$ et $(0, 2, 1)$.

Question 3 On peut toujours rendre le montant x uniquement avec des pièces c_m, c'est-à-dire avec x pièces. On en déduit la majoration $M(x) \leq x$.

D'autre part, si p est une représentation de x, alors $x \leq c_1 \sum_{i=1}^{m} p_i = c_1\|p\|$, car c_1 est le plus grand des c_i. On en déduit que $\|p\| \geq \dfrac{x}{c_1}$. Cependant, comme $\|p\|$ est entier, on a bien $\|p\| \geq \left\lceil \dfrac{x}{c_1} \right\rceil$. Ceci étant vrai pour toute représentation, c'est en particulier vrai pour une représentation de poids minimal.

Question 4 Notons e_i le vecteur de la base canonique de $\mathbb{R}^m$ dont seule la coordonnée i est non nulle et vaut 1. Soit $x \in \mathbb{N}^*$. Soit k le plus petit indice i tel que $c_i \leq x$.
Si p' est une représentation minimale de $x - c_i$, alors $p = p + e_i$ est clairement une représentation de x et donc on a $M(x) \leq \|p\| = \|p' + e_i\| = 1 + \|p'\| = 1 + M(x - c_j)$. On en déduit $M(x) \leq 1 + \min\limits_{k \leq i \leq m} M(x - c_i)$.
Réciproquement, si p est une représentation minimale de x, alors l'une des coordonnées $j \geq k$ de p est non nulle. De plus, la représentation $p - e_j$ est minimale pour $x - c_j$ (sinon, on pourrait construire une représentation de poids inférieur pour x), et donc $M(x) = \|p\| = 1 + \|p - e_j\| = 1 + M(x - c_j) \geq 1 + \min\limits_{k \leq i \leq m} M(x - c_i)$.

Question 5 On stocke les poids dans le tableau `poids`. La référence `k` me permet de garder en mémoire le plus petit indice i tel que $c_i \leq x$, sans avoir à le recalculer à chaque fois (k diminue forcément au fur et à mesure que y augmente). Dans la boucle, j'effectue une recherche de minimum.

```
let poids_min x c =
   let poids = Array.make (x + 1) 0 in
   let m = Array.length c in
   let k = ref (m - 1) in
   for y = 1 to x do
      if !k > 0 && c.(!k - 1) <= y then k := !k - 1;
      let mini = ref poids.(y - c.(m - 1)) in
      for i = !k to m - 1 do
         mini := min (!mini) poids.(y - c.(i))
      done;
      poids.(y) <- 1 + !mini
   done;
   poids;;
```

Question 6 De par les deux boucles imbriquées, on a une complexité en $O(xm)$. La complexité peut être moindre, notamment si k diminue peu.

Question 7 Il suffit de créer un tableau `pieces` tel que `pieces.(x)` garde en mémoire l'indice j tel que $M(x-c_j) = \min_{k \leq i \leq m} M(x-c_i)$. On reconstruit alors la représentation p à l'envers.

II Algorithme glouton et système canonique

Question 8 On crée un tableau qu'on modifiera à l'aide de la fonction auxiliaire `aux`. Cette fonction prend deux arguments : k qui représente l'indice à partir duquel on peut choisir des pièces et y qui représente le montant à rembourser. Si la pièce c_k est de valeur supérieure à y, alors on doit relancer un appel avec $k+1$. Sinon, on ajoute une pièce c_k et on relance avec $y - c_k$.

```
let glouton x c =
  let m = Array.length c in
  let p = Array.make m 0 in
  let rec aux k = function
    | 0 -> ()
    | y when c.(k) > y -> aux (k + 1) y
    | y -> p.(k) <- p.(k) + 1; aux k (y - c.(k)) in
  aux 0 x;
  p;;
```

Question 9 Notons $c = (a, 1)$ avec $a > 1$. Soit $x \geq 1$ et $x = aq + r$ sa division euclidienne par a. Alors l'algorithme glouton donnera la représentation (q, r) et $G(x) = q + r$.
Dès lors, si $p = (q', x - aq')$ est une autre représentation avec $q' < q$, alors $\|p\| - G(x) = q' + x - q' - q - r = q' - q + a(q - q') = (a-1)(q - q') \geq 0$.

Question 10 Par exemple $c = (4, 3, 1)$ et $x = 6$. On aura $\Gamma(x) = (1, 0, 2)$ mais $(0, 2, 0)$ est un contre-exemple.

Question 11 Raisonnons par l'absurde. Supposons x un contre-exemple et posons $\Gamma(x) = (g_1, \ldots, g_{n+1})$ et $p = (p_1, \ldots, p_{n+1})$ sa représentation minimale. Notons $k = \min\{i \in [\![1; n+1]\!], g_i \neq p_i\}$. Quitte à remplacer x par $x - \sum_{i=1}^{k-1} g_i e_i$ et à tronquer le système à gauche, supposons $k = 1$.
Comme $g_1 = \left\lfloor \frac{x}{q^n} \right\rfloor$, on a $(1 + g_1)q^n > x$ et donc $p_1 < g_1$. Quitte à remplacer x par $x - p_1 q^n$, on peut supposer $p_1 = 0$ et $g_1 \geq 1$.
Dès lors, remarquons que $\forall i, p_i \leq q - 1$. En effet, sinon $p' = (p_1, \ldots, p_{i-1} + 1, p_i - q, p_{i+1}, \ldots, p_{n+1})$ serait une représentation de x de poids strictement inférieur à p.
On a alors $x = \sum_{i=2}^{n+1} p_i q^{n+1-i} \leq (q-1)(1 + q + \ldots + q^{n-1}) = q^n - 1 < q^n \leq x$ car $q_1 \geq 1$. On conclut par l'absurde.

Question 12 Soit $x \geq 2$ et $p = (p_1, \ldots, p_8)$ une représentation minimale de x. On raisonne par étapes :

- Nécessairement, $p_8 \leq 1$, sinon $(p_1, \ldots, p_7 + 1, p_8 - 2)$ serait de poids inférieur.
- De même, $2p_7 + p_8 \leq 4$, sinon on a soit $p_7 = 2$ et $p_8 = 1$ et alors $(p_1, \ldots, p_6 + 1, 0, 0)$ est de poids inférieur, soit $p_7 \geq 3$ et alors $(p_1, \ldots, p_6 + 1, p_7 - 3, p_8 + 1)$ est de poids inférieur.
- De même, $5p_6 + 2p_7 + p_8 \leq 9$, sinon on a $5p_6 \geq 6$, soit $p_6 \geq 2$ et alors $(p_1, \ldots, p_5 + 1, p_6 - 2, p_7, p_8)$ est de poids inférieur.

De la même façon, on montre que pour $k \in [\![2; 8]\!]$, $\sum_{i=k}^{8} c_i p_i \leq c_{k-1} - 1$. Cela nous assure que p est bien la représentation donnée par l'algorithme glouton (si une somme est comprise entre c_{k-1} et c_k, alors il y aura nécessairement une pièce c_k dans la représentation minimale, car toutes les représentations minimales avec uniquement des pièces c_i pour $i > k$ sont de valeur inférieure à c_k).

Question 13 Pour $x = 48$, l'algorithme glouton donne $(1, 0, 1, 1, 0, 0)$ alors que $(0, 2, 0, 0, 0, 0)$ est de poids inférieur.

Question 14 Dans un premier temps, montrons que tout contre-exemple x vérifie $x > 1 + c_{m-2}$. En effet :

- Si $x = 1 + c_{m-2}$, alors $(0, \ldots, 0, 1, 0, 1)$ est une représentation de x, et elle est bien minimale, sauf si $x = c_{m-1}$, mais auquel cas, on aurait $\Gamma(x) = (0, \ldots, 1, 0, 0, 0)$ (qui est minimale).

- Si $x = c_{m-2}$, le résultat est immédiat.
- Si $x < c_{m-2}$, on se ramène à la question 9.

Remarquons alors que c_1+c_2 n'est pas un contre-exemple. Soit maintenant x le plus petit contre-exemple et supposons $x \geq c_1 + c_2 + 1$. Par hypothèse, on a $\forall y < x, G(y) = M(y)$. De plus, il existe une représentation p de x telle que $M(x) = \|p\| < G(x)$.
Si $p_1 \geq 1$, alors $G(x) = 1 + G(x - c_1)$, mais $x - c_1$ n'est pas un contre-exemple, donc $G(x - c_1) = M(x - c_1)$ et on conclut que $G(x) = 1 + M(x - c_1) = M(x)$ (absurde).
Dès lors, soit $k \geq 2$ tel que $p_k \geq 1$. On a $x > c_1 + c_2 \geq c_1 + c_k$. De même que précédemment, on a $G(x) = 1 + G(x - c_1) = 1 + M(x - c_1)$. Or, $M(x - c_1) \leq 1 + M(x - c_1 - c_k) = 1 + G(x - c_1 - c_k) = G(x - c_k) = M(x - c_k)$. On en déduit que $M(x - c_k) = M(x) - 1$, puis que $G(x) = 1 + M(x - c_1) \leq 1 + M(x - c_k) = M(x)$. x n'est donc pas un contre-exemple.

Question 15 La question précédente indique que le plus petit contre-exemple est à chercher entre $q+3$ et $2q$. On montre facilement que $2q$ est bien un contre exemple.
Dès lors, soit $x \in [\![q + 3; 2q - 1]\!]$. On a $\Gamma(x) = (1, 0, x - q - 1)$, de poids $x - q$. Soit p une représentation minimale de x. Si $p_1 = 1$, alors $p = \Gamma(x)$ nécessairement. Sinon, si $p_2 = 1$, alors $p = (0, 1, x - q)$, de poids $x - q + 1$, ou alors $p_2 = 0$ et alors $p = (0, 0, x)$, de poids x. Ces deux représentations étant de poids supérieur à $G(x)$, on en déduit que le plus petit contre-exemple est $2q$.

Question 16 Si c n'est pas canonique, alors son plus petit contre-exemple est supérieur ou égal à $f(q) + 2$. Si $x = f(q)+2)$, on a $\Gamma(x) = (1, 0, 2)$, de poids 3. Une représentation p de x telle que $p_1 = 1$ coïncide avec $\Gamma(x)$. Supposons alors $p = (0, p_2, p_3)$ une représentation de x. Si x est un contre-exemple, alors $p_2 + p_3 < 3$ et $p_2q + p_3 = f(q) + 2$. On distingue les cas :

- $p = (0, 0, 1)$, donc $x = 1$ et $f(q) = -1$: absurde.
- $p = (0, 0, 2)$, donc $x = 2$ et $f(q) = 0$: absurde.
- $p = (0, 1, 0)$, donc $x = q$ et $f(q) = q - 2 < q$: absurde.
- $p = (0, 1, 1)$, donc $x = q + 1$ et $f(q) = q - 1 < q$: absurde.
- $p = (0, 2, 0)$, donc $x = 2q$ et $f(q) = 2q - 2 > q$. C'est le seul cas possible.

Ainsi, $(2q - 2, q, 1)$ n'est pas canonique, avec pour contre-exemple $2q$.

Question 17 Les deux questions précédentes montrent que les bornes de la question 14 sont optimales. Ainsi, il suffit de tester les valeurs $x \in [\![c_{m-2} + 2; c_1 + c_2 - 1]\!]$ pour y trouver un contre-exemple. On utilise une question de poids, et on teste en utilisant les fonctions déjà écrites :

```
let norme p =
   let s = ref 0 in
   for i = 0 to Array.length p - 1 do
      s := !s + p.(i)
   done;
   !s;;

let canonique c =
   let m = Array.length c in
   if m <= 2 then true else begin
   let poids = poids_min (c.(0) + c.(1)) c in
   let b = ref true in
   for k = c.(m - 3) + 2 to c.(0) + c.(1) - 1 do
      if norme (glouton k c) > poids.(k) then b := false
   done;
   !b end;;
```

⋆⋆⋆