Option Informatique
Complexité

Sujet

21 octobre 2007

1 Décodage d'entiers

► On représente un entier n en base b pa rune chaîne de caractères. Par exemple $n = 125$ est représenté par `s :=»125` en base 10.

Question 1 *Écrire une fonction* `valeur s` *qui renvoie l'entier associé à* s*.*

Question 2 *Écrire une fonction* `compare s s'` *qui renvoie la valeur* `true` *quand* $s \geq s'$*. On ne calculera pas les entiers associés à* `s` *et* `s'` *car cette démarche est caduque si les entiers manipulés sont trop grands pour le type* `int`*.*

Question 3 *Évaluer la complexité de ces fonctions.*

2 Représentation d'un nombre en base b

► Soit n un entier naturel et b un entier nature avec $b \geq 2$. On se propose de calculer la représentation minimale en base b de n :

$$n = [t_{p-1}, t_0]_b$$

On rappelle l'algorithme :
- Soit u la suite définie par $u_0 = n$et pour $i \geq 0$, u_{i+1} est le quotient de la division euclidienne de u_i par b.
- p est le premier indice i tel que $u_i = 0$.
- t_i est le reste dans la division euclidienne de u_i par b.

Question 4 *a*

Donner une fonction `calcul_base_min` *telle que* `calcul_base_min n b` *affiche la représentation en base* b *de* n*.*

b Vérifier votre précédure à l'aide de quelques exemples en base 10.

c Soit n *un entier dont on connaît la réprésentation en base* b*. Comment fait-on pour connaître facilement le quotient et le reste dans la division euclidienne de* n *par* b *?*

Question 5 *Évaluer la complexité de la fonction précédente.*

3 Application : la multiplication rapide par un entier

► Soit n un entier et x un réel. On se proposer de calculer nx.

Question 6 *On pose* $u_n = nx$*. on a donc* $u_0 = 0$ *et pour tout* $n \geq 0,\ u_{n+1} = u_n + x$*.*

a Écrire une fonction `produit` *telle que* `produit n x` *renvoie* nx*, et qui ne fait que des additions.*

b Quelle est sa complexité ?

► On se propose d'écrire une nouvelle procédure plus compliquée, mais faisant moins d'additions. On considère l'écriture en base 2 de n :$n = [t_{p-1}, t_0]_2$.

On définit les suites y et v :
- $y_0 = x$ et pour $i \geq 0$: $y_{i+1} = 2y_i = y_i + y_i$
- $v_0 = 0$ et pour $i \geq 0$: $v_i + 1 = v_i + y_i$ si $t = 1$ et $v_{i+1} = v_i$ si $t = 0$

Question 7 *Montrer que* $v_p = nx$*.*

Question 8 *En déduire une fonction* `produit_rapide` *telle que* `produit_rapide n x` *renvoie* nx *et qui ne fait que des additions. On calculera* t_i *au fur et à mesure du calcul des* y_i *et* v_i*.*

Question 9 *Evaluer la complexité de cette fonction.*

4 Calcul du PGCD

► La suite de Fibonnaci est définie par $f_0 = f_1 = 1$ et $\forall n \geq 2,\ f_n = f_{n-1} + f_{n-1}$.

Question 10 *Donner une fonction* `calcul_fibo` *qui vous permette d'obtenir le cinquantième terme de la suite de Fibonacci en moins de vingt secondes.*

Question 11 *Évaluer la complexité de cette fonction.*

► Soient a et b deux entiers naturels avec $a \geq b$ On veut écrire une fonction entière `pgcd(a,b)` qui renvoie le PGCD de a et b. On impose une contrainte : utiliser la fonction `n mod p` qui renvoie le reste dans la division euclidienne de `n` par `p`.

On introduit l'environnement suivant : Env $= n, p$: `int`, et la propriété $H =$"$n \geq p \geq 0$ et $n \wedge p = a \wedge b$".

Question 12 *a Quelle valeur suffit-il de donner initialement à n et p pour avoir H vraie ?*

b On suppose $p = 0$. Quel est, dans chaque cas, le PGCD de n et p ?

c On suppose que p est un entier strictement positif. Comment mettre à jour n et p afin de se rapprocher du cas d'arrêt ?

► On sait que si $n \geq p$ alors $n \wedge p = p \wedge (n \mathrm{mod} p)$. en déduire une modification de l'environnement permettant de se *rapprocher* de la condition $p = 0$ et laissant la propriété H invariante.

Question 13 *En déduire une écriture itérative puis récursive multiple de la fonction* `pgcd(a,b)`*. Justifier de son arrêt et de sa validité.*

Question 14 *On étudie dans cette question la version récursive. On introduit la suite de Fibonnaci (f_n)<>.*

a On suppose que $a \geq b > 0$ Établir que si `pgcd(a,b)` *fait n appels récursifs avec $n \geq 1$ alors $a \geq f_n$ et $b \geq f_{n-1}$.*

b Montrer que $f_n \geq (\frac{1+\sqrt{5}}{2})^{n-1}$

c En déduire la complexité de la version récursive est en $O(\ln b)$. Que dire de la complexité de la version itérative ?

5 Algorithme d'Euclide étendu

Fonction gcde(a, b : **entier**) :
entier $\times$ **entier** $\times$ **entier**
    M ← (a,1,0)
    N ← (b,0,1)
    Tant que $(N[1] \neq 0)$ **faire**
        q ← quotient (M[1],N[1])
        r← reste(M[1],N[1])
        tmp← N
        N← (r,$M[2]-q\times N[2]$,$M[3]-q\times N[3]$)
        M←tmp
    Fait
    Retourner M
Fin

► Notons $(M_k)_{O \leq k \leq n}$ la suite des triplets définis par l'algorithme précédent, avec M_{n+1} le premier triplet tel que $M_{n+1}[1] = 0$. Pour $0 \leq k \leq n$, nous notons $r_k = M_k[1]$, $u_k = M_k[2]$, $v_k = M_k[3]$.

Question 15 *1. Montrer que pout tout $O \leq k \leq n$ nous avons $au_k + bv_k = r_k$*

2. Montrer que pour un $k \geq 0$ nous avons $u_k \geq 0$ et $u_{k+1} \leq 0$

Question 16 *Que renvoie l'algorithme* `egcd` *? Justifier sa validité.*

Question 17 *Implémenter cette fonction en Caml, de façon récursive, et évaluer sa complexité.*