

Option Informatique

Complexité et arbres

Sujet

20 octobre 2006

1 Le retour de Fibonacci

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

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

Question 1 Quelle valeur suffit-il de donner initialement à n et p pour avoir H vraie ?

Question 2 On suppose $p = 0$. Quel est, dans chaque cas, le PGCD de n et p ?

► On suppose que p est un entier strictement positif.

Question 3 On sait que si $n \geq p$ alors $n \wedge p = p \wedge (n \bmod p)$. en déduire une modification de l'environnement permettant de se rapprocher de la condition $p = 0$ et laissant la propriété H invariante.

2 Parcours en largeur d'abord

► Les parcours *préfixe*, *infixe* et *suffixe* sont dits parcours en *profondeur d'abord* : on descend le long d'une branche tant que c'est possible. On veut à présent faire un parcours d'arbre en *largeur d'abord*. Durant un tel parcours, on traite la racine, puis ses fils, puis les fils de ces fils, etc.

On se place dans le cas des arbres binaires, dont les feuilles sont étiquetées par \mathbb{N} et les noeuds internes

Question 4 En déduire une écriture itérative puis récursive multiple de la fonction $\text{pgcd}(a, b)$. Justifier de son arrêt et de sa validité.

► On étudie dans cette question la version récursive. On introduit la suite de Fibonacci (f_n) définie par $f_0 = f_1 = 1$ et la relation de récurrence $\forall n \geq 2, f_n = f_{n-1} + f_{n-2}$.

Question 5 On suppose que $a \geq b > 0$. Établir que si $\text{pgcd}(a, b)$ fait n appels récursifs avec $n \geq 1$ alors $a \geq f_n$ et $b \geq f_{n-1}$.

Question 6 Montrer que $f_n \geq \left(\frac{1+\sqrt{5}}{2}\right)^{n-1}$

Question 7 En déduire la complexité de la version récursive est en $O(\ln b)$. Que dire de la complexité de la version itérative ?

étiquetés par \mathbb{R} .

Question 8 Donner le type CAML d'un arbre binaire correspondant à cette définition.

► On notera fonc_f et fonc_n les fonctions à appliquer respectivement aux feuilles et aux noeuds de l'arbre. On notera T l'arbre initial à traiter. On utilise

une liste d'arbres que l'on notera L : L doit contenir à chaque étape la liste des sous-arbres qui restent à traiter.

Question 9 *Que doit valoir L initialement ?*

► On suppose que L contient la liste des sous-arbres qui restent à traiter. On isole la tête de L : c'est l'arbre qui doit être traité. On le note (G, x, D) .

Question 10 *Que doit-on faire pour le traitement, la réactualisation de L et la poursuite du calcul en largeur d'abord ? Proposer une structure plus adéquate pour L qu'une liste.*

Question 11 *Quand arrête-t-on le traitement ?*

Question 12 *En déduire une fonction `traite_L` `fonc_f` `fonc_n` qui effectue le traitement en largeur d'abord des arbres de L .*

Question 13 *En déduire une fonction `largeur` `fonc_f` `fonc_n` T qui effectue le traitement en largeur d'abord de T .*

Question 14 *Tester votre fonction en prenant pour `fonc_f` et `fonc_n` des simples fonctions d'affichage, de façon à imprimer les étiquettes des noeuds de l'arbre en largeur d'abord.*

Question 15 *Quelle est la complexité de la fonction `largeur`, en nombre d'opérations élémentaires sur la liste L (insertion, lecture en tête), exprimée en fonction du nombre n de noeuds de T ? Quelle serait sa complexité si la liste L était remplacée par une file de priorité (où l'insertion en queue est faite en temps constant) ?*

3 Parcours en largeur et représentation contiguë

► Dans cette section, on supposera que les noeuds de l'arbre T sont étiquetés par des objets de type `'n` quelconque.

Question 16 *Donner le type CAML `'n` arbre correspondant à cette définition.*

► On suppose maintenant qu'on dispose d'un arbre binaire quasi-complet, c'est à dire un arbre dont tous les étages sont remplis, sauf le dernier, qui est «tassé» vers la gauche. Notons que l'arbre T n'est ainsi plus nécessairement strict : tous les noeuds internes ont deux fils, sauf le noeud interne le plus à droite

Question 17 *Donner un encadrement du nombre de noeuds n d'un arbre quasi-complet en fonction de sa hauteur h .*

Question 18 *Exprimer le rang du k -ième sommet du d -ième étage dans un parcours en largeur d'abord. Montrer qu'il y a $i - 1$ sommets entre un sommet d'indice i et son fils gauche.*

► On suppose donné un tableau D donnant le parcours en largeur d'abord d'un arbre.

Question 19 *Quel est l'indice des fils (gauche, droit) du sommet d'indice i ? Quel est celui de son parent ?*

Question 20 *Étant donné le tableau D représentant le parcours en largeur d'un arbre T , écrire une fonction `tree_of_array` qui renvoie l'arbre T , de type `'n` arbre.*