Option Informatique
Complexité et arbres

Sujet

19 octobre 2007

1 Parcours en escalier dans un tableau.

(d'après ESIM 2000)

► Soit M une matrice de $n+l$ lignes et de $n+l$ colonnes ($n > 0$) telle que $m_{i,j} = 2^i 3^j$ pour $0 \leq i \leq n$ et $0 \leq j \leq n$.

Question 1 *Décrire un algorithme de remplissage du tableau à deux dimensions représentant la matrice M.*

Question 2 *Quel est le nombre total de multiplications de votre algorithme ?*

Question 3 *Ecrlre la fonction* `pgcd` *qui a prend pour arguments une matrice M (telle que $m_{i,j} = 2^i 3^j$ pour $0 \leq i \leq n$ et $0 \leq j \leq n$) et quatre indices i, j, k, l dans $[0, n]$ et retourne pour résultat la valeur du plus grand commun diviseur de $m_{i,j}$ et de $m_{k,l}$.*

Question 4 *On dispose d'une matrice M telle que $m_{i,j} = 2^i 3^j$ pour $0 \leq i \leq n$ et $0 \leq j \leq n$, décrire un algorithme qui étant donné un entier p,$1 \leq p \leq 2^n 3^n$, calcule les indices i, j du plus grand des minorants de p de la forme $2^i 3^j$. On s'efforcera dans cette question de minimiser le nombre de comparaisons en exploitant la remarque qui suit.*

► *Remarque* : Si sur la ligne i_0 le plus grand des minorants de p se trouve en colonne j_0 , alors sur la ligne $i_0 + 1$ le plus grand des minorants de p est :

- Soit I'élément qui se trouve sur en ($i_0 + 1, j_{,0}$), si $m_{i_0+1,i_0} = 2m_{i_0,j_0}$ est inferieur ou égal à p.
- Soit I'élément qui se trouve en ($i_0 + 1, j_0 - 1$) , si $m_{i_0+1,i_0} = 2m_{i_0,j_0}$ est strictement supérieur à p, car dans ce cas
  $m_{i_0+1,i_0-1} = \frac{2}{3} m_{i_0,j_0}$ est alors le plus grand des minorants de p sur la ligne $i_0 + 1$.

Question 5 *Combien de comparaisons mettant en jeu un éliment de la matrice M effectue votre algorithme ?*

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 noeud internes étiquetés par $\mathbb{R}$.

Question 6 *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 7 *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 8 *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 9 *Quand arrête-t-on le traitement ?*

Question 10 *En déduire une fonction* `traite_L fonc_f fonc_n` *qui effectue le traitement en largeur d'abord des arbres de L.*

Question 11 *En déduire une fonction* `largeur fonc_f fonc_n T` *qui effectue le traitement en largeur d'abord de T.*

Question 12 *Tester votre fonction en prenant pour* `fonc_f` *et* `fonc_n` *des simples foncitons d'affichage, de façon à imprimer les étiquettes des noeuds de l'arbre en largeur d'abord.*

Question 13 *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 contigüe

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

Question 14 *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 15 *Donner un encadrement du nombre de noeuds n d'un arbre quasi-complet en fonction de sa hauteur h.*

Question 16 *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 17 *Quel est l'indice des fils (gauche, droit) du sommet d'indice i ? Quel est celui de son parent ?*

Question 18 *Étant donné le tableau D représentant le parcour sen largeur d'un rabre T, écrire une fonction* `tree_of_array` *qui renvoie l'arbre T, de type* `'n arbre`.