Université de Marne-la-Vallée

Licence d'informatique et d'I.U.P.
Examen d'algorithmique

Janvier 2001, durée : 2h

Seule la consultation des supports de cours, travaux dirigés et travaux pratiques est autorisée pendant le contrôle.

Arbres feuillus

Les arbres considérés dans cet exercice sont des arbres binaires feuillus (complets) dont les nœuds sont des entiers. À un tel arbre A on associe le mot $Str(A)$ sur l'alphabet $\{\mathtt{0},\mathtt{1}\}$ défini par :

$$Str(A) = \begin{cases} \mathtt{1} & \text{si } A \text{ est réduit à une feuille,} \\ \mathtt{0}\,Str(G)\,Str(D) & \text{si } A = (r,G,D), \end{cases}$$

où r est la racine de A, G son sous-arbre gauche et D son sous-arbre droit.

1. Calculer $Str((1,(2,(3),(4)),(5)))$ et $Str((1,(2),(3,(4),(5))))$.
   Dessiner les arbres qui correspondent aux mots `0001111` et `0101011`.
2. Soient $x = Str(A)$ et $y = Str(B)$ pour deux arbres A et B.
   Montrer que $x = y$ si et seulement si A et B sont égaux à la numérotation près de leurs nœuds (autrement dit, s'il existe une bijection entre leurs nœuds qui respecte la structure d'arbre).
3. Indiquer comment mémoriser dans un fichier le code obtenu par la méthode de Huffman, en adaptant la fonction Str à appliquer à l'arbre de Huffman du codage.
4. Écrire une fonction C qui produit sur la sortie standard $x = Str(A)$ pour un arbre A.
5. Écrire une fonction C qui, à partir d'un mot binaire $Str(A)$ sur l'entrée standard, reconstruit l'arbre A.

6. Quel est le temps d'exécution de chacune des fonctions que vous avez obtenues aux questions 4 et 5 ?

Sous-suites croissantes

Le but de cet exercice est d'écrire un algorithme de calcul de la longueur maximale des sous-suites croissantes d'une suite d'entiers. Étant données deux suites d'entiers $U = (U[0],U[1],\dots,U[k-1])$ et $X = (X[0],X[1],\dots,X[m-1])$ avec $k > 0$ et $m > 0$, on dit que U est une **montée** de X s'il existe des indices $i_0, i_1, \dots, i_{k-1}$ tels que

$$\begin{cases} 0 \leq i_0 < i_1 < \cdots < i_{k-1} < m, \\ U[0] = X[i_0],\ U[1] = X[i_1],\ \dots,\ U[k-1] = X[i_{k-1}] \quad \text{et} \\ U[0] \leq U[1] \leq \cdots \leq U[k-1]. \end{cases}$$

On note $LSC(X)$ la longueur maximale des montées d'une suite d'entiers non vide X.

Soit k un entier naturel, $0 < k \leq LSC(X)$. On dit d'une sous-suite V de longueur k de X qu'elle est une **meilleure montée** si son dernier élément est inférieur (ou égal) à chacun des derniers éléments des autres montées de longueur k. On note alors $T_X[k]$ le dernier élément d'une meilleure montée V de longueur k de X (c'est-à-dire, $T_X[k] = V[k-1]$).

Pour $A = (6,2,4,0,5,3)$, on a $LSC(A) = 3$, $T_A[1] = 0$, $T_A[2] = 3$, et $T_A[3] = 5$.

7. Calculer $LSC(Z)$ et T_Z pour la suite $Z = (9,3,1,5,0,7,11,2,7,4,3)$.

Soient X une suite d'entiers non vide et $\ell = LSC(X)$.

8. Montrer les inégalités $T_X[1] \leq T_X[2] \leq \cdots \leq T_X[\ell]$.

Soient y un entier et $Y = (X[0],X[1],\dots,X[m-1],y)$.

9. Exprimer $LSC(Y)$ et T_Y en fonction de $\ell = LSC(X)$ et T_X lorsque $T_X[i-1] \leq y < T_X[i]$, pour un entier i tel que $1 < i \leq \ell$. Que dire si $y < T_X[1]$? Que dire si $T_X[\ell] \leq y$?
10. Indiquer comment calculer $LSC(X)$ et T_X en utilisant l'itération de la question 9.

    Quel est le temps d'exécution de votre méthode?