Esiee-Paris - cours d'algorithmique - TP 2 - 17 mars 2015

R. Natowicz, I. Alamé, A. Çela, D. Courivaud

1. Fibonacci dichotomique. On définit $F(k)$, vecteur de couples de valeurs consécutives de la fonction de Fibonacci, par $F(k) = \begin{bmatrix} \text{fibo}(k) \\ \text{fibo}(k-1) \end{bmatrix}$; en particulier, $F(1) = \begin{bmatrix} \text{fibo}(1) \\ \text{fibo}(0) \end{bmatrix}$.

Étant donnée la matrice $M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, pour tout $k, k \geq 2$, on a $F(k) = M \times F(k-1)$.

Ainsi : $F(2) = M \times F(1)$, $F(3) = M \times F(2) = M^2 \times F(1)$, ..., $F(n) = M^{n-1} \times F(1)$.

Pour calculer la valeur fibo(n) on peut donc calculer la matrice M^{n-1} puis retourner la première composante du vecteur $M^{n-1} \times F(1)$, c'est-à-dire la valeur $M^{n-1}[0][0]$.

1. Écrire une fonction `int[][] produit(int[][] X, int[][] Y)` qui retourne le produit P des deux matrices X et Y. Rappel : la matrice produit P est de terme général $P[i][j] = \sum_k x_{ik} y_{kj}$. Inutile de donner la propriété invariante. Remarque : si les deux matrices ne sont pas compatibles pour le produit matriciel, votre fonction affichera un message d'erreur et retournera une matrice vide (en Java : la valeur `null`.)

2. Écrire une fonction `int[][] puissance(int[][] A, int b)` qui retourne la matrice A^b par un calcul dichotomique, où A est une matrice carrée et b un entier positif ou nul. Donner la propriété invariante de votre programme et commenter la fonction avec cette propriété. Vous aurez besoin d'une fonction `int[][] copie(int[][] M)` qui retourne une copie de la matrice M, et d'une fonction `int[][] unite(int n)` qui retourne la matrice unité de dimension n (matrice carrée $n \times n$). Si la matrice A n'est pas carrée, votre fonction affichera un message d'erreur et retournera une matrice vide.

3. écrire la fonction `fibo(int n)` qui retourne, en $\Theta(\log n)$ multiplications, la valeur fibo(n).

2. Premier plus long sous-tableau constant (pplstc). Soit $T[0:n]$ un tableau d'entiers. Calculer en complexité $\Theta(n)$ les indices d et f du pplstc de T. Exemple : le pplstc de $T[0:6] = [2, 1, 1, 3, 4, 4]$ est $T[1:3] = [1, 1]$. Donner la propriété invariante de votre programme et commenter la fonction avec cette propriété. Votre fonction retournera son résultat dans un tableau `int[]df` de longueur 2, tel que `df[0]=d` et `df[1]=f`. Sur l'exemple ci-dessus votre fonction retournera `int[]df={1,3}`.

3. Sous-tableau de somme maximum. Soit $T[0:n]$ un tableau d'entiers. Calculer en complexité $\Theta(n)$ les indices d et f, $d < f$, du premier sous-tableau $T[d:f]$ de somme maximum. Exemple : avec $T[0:10] = [-2, 3, -1, 5, -2, 4, -3, -6, 8, 1]$, le premier sous-tableau non vide de somme maximum est $T[d:f] = T[1:6] = [3, -1, 5, -2, 4]$, de somme $s = 9$. Donner la propriété invariante et commenter la fonction avec cette propriété. Votre fonction retournera son résultat dans un tableau `int[]dfs` de longueur 3, tel que `dfs[0]=d`, `dfs[1]=f`, et `dfs[2]=s`. Sur l'exemple ci-dessus votre fonction retournera `int[]dfs={1,6,9}`.

4. Quelques programmes récursifs.

1. La valeur minimum d'un sous-tableau vide est infinie, la valeur minimum d'un sous-tableau de longueur un est l'unique valeur du sous-tableau. Remarque : en Java, l'entier infini est le plus grand entier représentable, `int infini = Integer.MAX_VALUE`.

   La fonction `int[]min(int[]T){int n=T.length; return min(T,0,n);}` retourne la valeur minimum du tableau `T[0:n]` par l'appel principal `min(T,0,n)`.
   Écrire la fonction `int[]min(int[]T, int i ,int j)` qui retourne la valeur minimum du sous-tableau `T[i:j]` par une approche diviser pour régner ;

2. la fonction `void endroit(int[]T){e(T, 0);System.out.println();}` affiche sur une ligne les valeurs du tableau `T[0:n]` puis affiche un retour à la ligne.
   Écrire la fonction `e(int[]T, int i)` qui affiche sur une ligne les valeurs du sous-tableau `T[i:n]`.

3. la fonction `void envers(int[]T){r(T, 0);System.out.println();}` affiche sur une ligne les valeurs du tableau `T[0:n]` dans l'ordre inverse, `T[n-1], ..., T[0]`, puis affiche un retour à la ligne.
   Écrire la fonction `r(T, i)` qui affiche à l'envers sur une ligne les valeurs du tableau `T[i:n]`. On n'autorise que l'accès au premier élément du tableau : `System.out.print(T[i] + " ")`.

Une classe TP2 est jointe à ce sujet. Complétez cette classe TP2 pour en faire votre rapport. Les TP sont à faire en binôme. Les rapports sont à rendre au début de la prochaine séance de TP (lundi 30 mars ou mardi 1e avril.) Rapports courts (3 pages maximum), avec exemples d'exécutions.