Coupes d'un tableau

Laure Danthony
`http://www.ens-lyon.fr/~ldanthon/`
D'après ENS Ulm/Lyon MP 2002

Introduction et énoncé du problème

L'énoncé initial constituait un exemple de problème pratique d'algorithmique et de programmation, devant machine, du concours 2002 des ENS Ulm et Lyon. Ce sujet avait été proposé en mai 2002 aux préparationnaires de ce concours, car l'épreuve était nouvelle. On ne fournit pas ici de squelette, à vous de bien gérer votre feuille de travail.

Le sujet propose divers parcours d'un tableau T de taille $n = 1000$ dont tous les éléments valent 0 ou 1. Le tableau est initialisée comme suit. On définit tout d'abord la suite récurrente d'entiers suivante :
- $x_0 = 13$
- $x_i = (x_{i-1} \times x_{i-1})$ modulo 3953 pour i compris entre 1 et n.

Ensuite, pour tout $1 \leq i \leq n$, on remplit le tableau par $T[i] := x_i$ modulo 2.

1 Visualisation, initialisation, vérifications

L'objet de ce paragraphe est d'écrire quelques fonctions bien utiles pour la suite, notamment pour la visualisation des résultats.

► Exo :
- Préparer votre feuille de travail, déclarer notamment n comme une constante, le type `table`, ...
- Ecrire une `procedure visu(t:table,i,j)` qui imprime à l'écran le sous-tableau `t[i..j]`.
- Déclarer le tableau T de travail, et l'initialiser avec les bonnes valeurs.
- Combien d'éléments du tableau sont égaux à 1 ? Quel est l'indice du dernier d'entre eux ? On pourra judicieusement écrire une procédure `question1(t:table)` qui réalise ce travail sur un tableau quelconque, puis l'appeler sur le tableau T. (réponses 4715 et 10000)

2 Notion de coupe, palindromes

Une **coupe** $C[i, j[$ du tableau T est une suite d'éléments consécutifs $T[i]$, $T[i + 1]$, $\ldots T[j - 1]$ (un "sous-tableau"), où $1 \leq i \leq j \leq n - 1$. Noter que $C[i, j[$ est de taille $j - i$.

Une coupe $C[i, j[$ est un **palindrome** si elle est égale à la coupe obtenue en prenant ses éléments à l'envers, c'est-à-dire si $T[i] = T[j-1]$, $T[i+1] = T[j-2]$, $T[i+2] = t[j-3]$, ...

► On cherche dans un premier temps à déterminer si il existe dans T des coupes de longueur 7 qui sont des palindromes :

- Ecrire un algorithme qui effectue ce travail, le tester sur le tableau T.
- Evaluer le temps d'exécution de votre algorithme (en nombre de tests) en fonction de n la taille du tableau considéré.
- Modifier la fonction (ou procédure) précédente pour qu'elle imprime le deuxième plus petit indice i tel que $C[i, j[$ soit un palindrome de longueur 7. (réponse 999 palindromes, le deuxième à la position 20).

► On cherche à déterminer la longueur de la plus longue coupe de T qui soit un palindrome.

- Ecrire une fonction (ou procédure) qui effectue ce travail et vérifier que la réponse est 18.
- Modifier la fonction/procédure précédente qui imprime en plus le nombre de tels palindromes et la position du deuxième. (réponses : 71 et 205).

3 Coupes nulles, coupes équilibrées

On cherche à déterminer la longueur maximale d'une coupe constituée uniquement d'éléments nuls, et l'indice de la première de ces coupes.

► Exo :

- Ecrire une fonction/procédure qui résout le problème. (réponses : longueur max : 8 et position 47)
- La tester. Evaluer sa complexité.
- Quels améliorations peut-on apporter à cet algorithme ?

Une coupe est dite **équilibrée** si elle comprend autant de 0 que de 1. On cherche à déterminer la longueur maximale d'une coupe équilibrée, combien il y a de telles coupes, ainsi que l'indice de la deuxième.

► Mêmes questions que précédemment. (réponses 132, 71 et 195)