Parcours de tableau
Épreuve pratique d'algorithmique et de programmation

(Version beta, testée partiellement)

Alain Darte et Yves Robert
[Alain.Darte,Yves.Robert]@ens-lyon.fr

Mai 2002

Ce document constitue une épreuve type pour l'épreuve pratique d'algorithmique et de programmation du concours des ENS Ulm et Lyon, informatique. Le texte indenté, en caractères gras, est fait de commentaires expliquant l'esprit du sujet. Le reste du texte est le sujet proprement dit.

> **L'épreuve a une durée de quatre heures. En 2002, elle se déroulera à l'ENS Lyon dans une salle informatique dotée d'une quinzaine de postes de travail de type PC. Ces postes de travail ne seront pas connectés au réseau.**
>
> **Une douzaine de candidats passe l'épreuve simultanément, chacun devant un poste de travail. Au début de l'épreuve, les candidats reçoivent un sujet. Le sujet est identique pour tous les candidats d'une même séance ; il y a évidemment un sujet différent pour chaque séance de l'épreuve. À la fin de l'épreuve, les candidats rendent une copie papier, rédigée à la main, sur laquelle ils ont noté les réponses aux questions posées dans le sujet.**
>
> **Comme son nom l'indique, l'épreuve est de nature algorithmique, et s'accompagne d'une mise en oeuvre concrète sur machine. Les questions ont un double but : évaluer la solution algorithmique proposée par le candidat pour résoudre le problème posé, et vérifier la justesse et l'efficacité de la mise en oeuvre réalisée. On ne demandera pas les sources des programmes réalisés pour répondre aux questions.**

Le sujet propose divers parcours d'un tableau T de taille $n = 10000$ dont tous les éléments valent 0 ou 1. Le tableau est initialisé comme suit. On définit la suite récurrente d'entiers $(x_i)_{0 \leq i \leq n}$ par :

- x_0 est égal au numéro inscrit sur votre table d'examen,
- $x_i = (x_{i-1} \times x_{i-1})$ modulo 3953 pour i compris entre 1 et n.

Puis, pour $1 \leq i \leq n$, on pose $T[i] = x_i$ modulo 2.

> **L'énoncé pourra imposer tout autre principe de génération des données. L'important n'est pas que la séquence soit vraiment aléatoire, mais que (a) le candidat et le correcteur (qui contrôlera la justesse des programmes non pas par leur texte mais par les réponses qu'ils fourniront) travaillent tous deux sur les mêmes données et (b) qu'il soit difficile, voire impossible, de répondre aux questions sans un programme.** ATTENTION DONC DE NE PAS SE TROMPER LORS DES INITIALISATIONS ! **Néanmoins, le candidat pourra être amené, pour la mise au point de son programme, à considérer des petites valeurs de n ou même une initialisation particulière du tableau pour vérifier "à la main" si son programme fournit les bonnes réponses.**

Question 1. *Combien d'éléments du tableau sont égaux à 1 ? Quel est l'indice de l'avant-dernier d'entre eux ?*

Une **coupe** $C[i..j[$ du tableau est une suite d'éléments consécutifs $T[i]$, $T[i+1]$, …, $T[j-1]$, où $1 \leq i \leq j \leq n+1$. Noter que $C[i..j[$ est de longueur $j - i$ (si $j = i$, la coupe est vide).

Question 2. *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 $T[i] = T[j-1]$, $T[i+1] = T[j-2]$, $T[i+2] = T[j-3]$, etc. On cherche à déterminer s'il existe de coupes de longueur 7 qui sont des palindromes.*

1. *Expliquer brièvement l'algorithme que vous mettez en oeuvre.*

   On attend ici que le candidat explique les principes de son algorithme sans fournir dans son rapport le code complet de son programme. Ici, le candidat pourra expliquer qu'il mettra en place deux boucles imbriquées, l'une sur i, l'une sur j, et que pour chaque couple (i, j), il vérifiera si la coupe $C[i..j[$ est un palindrome (par une troisième boucle) et si celui-ci est de longueur 7. De plus, un compteur permettra de donner le nombre de palindromes identifiés. Un pseudo-code pourra être fourni pour illustrer ce propos. Bien sûr, ici, puisqu'on cherche des palindromes de longueur 7 exactement, une seule boucle comprenant 3 tests est suffisante puisqu'il suffit de considérer l'indice $j = i + 7$ pour un i donné. La première réponse est juste également, quoique moins satisfaisante. Le correcteur en tiendra compte.

2. *Définir parmi les opérations élémentaires effectuées par votre programme celles qui sont les plus significatives (justifier votre choix) et indiquer leur nombre en fonction de n, la taille du tableau T.*

   On attend ici que la candidat précise par exemple qu'il compte le nombre de comparaisons entre éléments du tableau effectuées. Ici, avec la première approche, le candidat devrait répondre $O(n^3)$ (en justifiant) ou s'il souhaite être plus précis, $\sum_{i=1}^{n}\sum_{j=i}^{n+1}\lceil\frac{j-i}{2}\rceil$) (expression qu'il n'essaierait pas de préciser, sauf si l'énoncé le demandait explicitement). Avec la seconde approche, le candidat pourrait répondre $O(n)$ ou même $3(n-6)$ s'il souhaite être plus précis. Le candidat pourra également mettre en place un compteur dans son programme qui fournira la valeur exacte due à son implantation.

3. *Donner le nombre de coupes de longueur 7 qui sont des palindromes dans le tableau et le deuxième plus petit indice i tel que $C[i..j[$ soit un palindrome de longueur 7.*

   On attend ici une réponse succincte, simplement le résultat du programme. Par exemple ici, si $x_0 = 13$, la réponse est 999 et le deuxième palindrome est en position 20. Le candidat peut fournir le palindrome en question pour être sûr que le correcteur a bien compris.

Question 3. *On cherche à déterminer la longueur de la plus longue coupe qui soit un palindrome. Répondre aux mêmes questions que précédemment pour ce nouveau problème.*

Le candidat doit reprendre le même schéma d'explications que pour le problème de la recherche d'un palindrome de longueur 7. Surtout ne pas oublier de donner les réponses de type "valeur numérique" qui seules permettent au correcteur de vérifier que le programme fonctionne correctement. Ici, toujours pour $x_0 = 13$, la longueur maximale d'un palindrome est 18, il y en a 71 et le second figure en position 205.

Question 4. *On cherche à déterminer la longueur maximale d'une coupe composée uniquement d'éléments nuls, et combien il y a de telles coupes dans le tableau.*

1. *Répondre aux mêmes questions que précédemment pour ce nouveau problème.*
2. *Quelles améliorations pourriez-vous apporter à votre algorithme ?*

   On cherche ici à ce que le candidat, après avoir éventuellement proposé une première solution correcte mais non optimisée, se rende compte qu'un autre algorithme, plus rapide, est possible. Ainsi, ici, une première approche comportant deux boucles imbriquées est possible, mais avec un simple parcours du tableau (une seule boucle), on peut répondre à la question. On attend une explication des principes à mettre en oeuvre et une nouvelle justification quant au nombre d'opérations effectuées. En revanche, on ne demande aucune programmation supplémentaire.

Question 5. *Une coupe est* **équilibrée** *si elle comprend autant de* 0 *que de* 1. *On cherche à déterminer la longueur maximale d'une coupe équilibrée, et combien il y a de telles coupes. Mêmes questions que précédemment.*

> **Ne pas oublier de fournir les résultats du programme et de discuter des améliorations possibles. De nombreuses solutions sont possibles.**

Question 6. *Une coupe est* **répétée** *si elle est la concaténation de deux coupes identiques. On cherche à déterminer la longueur maximale d'une coupe répétée et combien il y a de telles coupes. Mêmes questions que précédemment.*

Question 7. *On cherche à déterminer la suite de* 0 *et de* 1 *qui soit à la fois une coupe du tableau* T *et une coupe du tableau* U *obtenu en considérant les éléments de* T *en ordre inverse, c'est-à-dire le tableau* U *tel que* $U[i] = T[10000 - i + 1]$. *Mêmes questions que précédemment. Même question en supposant simplement que la suite de* 0 *et de* 1 *soit contenue (dans l'ordre, mais pas nécessairement consécutivement comme pour une coupe) à la fois dans* T *et dans* U.

Question 8. *On interprète à présent une coupe comme l'écriture en base* 2 *d'un entier. On cherche à déterminer deux coupes disjointes de même longueur dont la somme (avec l'interprétation en base* 2*) soit maximale et soit une coupe du tableau. Mêmes questions que précédemment. Préciser la somme obtenue par les indices des coupes correspondantes dans le tableau.*

> **Remarque générale : les correcteurs n'attendent pas nécessairement que les candidats parviennent à répondre à toutes les questions dans le temps imparti.**