Comptage de mots reconnus par un automate.

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles normales supérieures

Durée de l'épreuve : 3 heures 30 minutes

Juin/Juillet 2021

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Il vous a été donné un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion !). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures !

Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple : $O(n^2)$, $O(n \log n)$,...

Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***. Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

1 Préliminaires

Notations et rappels sur les automates et les mots

On rappelle que pour deux entiers naturels a et b, $a \mod b$ désigne le reste de la division entière de a par b, c'est à dire l'unique entier r avec $0 \leq r < b$ tel que $a = k \times b + r$ pour $k \in \mathbb{N}$.
Un mot w de longueur n sur un alphabet Σ est constitué de n lettres $w_0 \dots w_{n-1}$ (toutes dans Σ).
Étant donné un mot $w = w_0 \dots w_{n-1}$ de longueur n et deux entiers $0 \leq i \leq j \leq n$, on note $w[i : j]$ le mot $w_i \dots w_{j-1}$, c'est à dire le facteur du mot w entre les positions i (incluse) et j (exclue).
Un automate fini déterministe est usuellement décrit par un quintuplet $(\Sigma, \mathcal{Q}, q_0, \delta, \mathcal{F})$ où :

- Σ représente l'alphabet sur lequel l'automate travaille,
- $\mathcal{Q}$ représente l'ensemble des états,
- q_0 est l'unique état initial,
- δ est la fonction de transition de l'automate, telle que $\delta(q, c)$ donne l'état dans lequel l'automate se trouve après avoir lu la lettre c dans l'état q,
- $\mathcal{F}$ est l'ensemble des états finaux.

Dans ce sujet, Σ et $\mathcal{Q}$ seront toujours des ensembles de nombres entiers de la forme $0, 1, 2 \dots, n-1$ (où $n > 0$ est la taille de l'ensemble) tandis que l'état initial sera toujours 0. Ainsi Σ et $\mathcal{Q}$ seront entièrement définis par leur taille et donc, pour ce sujet, l'automate est entièrement décrit par le quadruplet $(L, Q, \delta, \mathcal{F})$ où L est la taille de l'alphabet, Q est le nombre d'états de l'automate, δ est la fonction de transition et $\mathcal{F}$ est l'ensemble des états finaux.
De façon usuelle, on étend la fonction de transition δ aux mots de la façon suivante $\delta(q, \epsilon) = q$ (ici ϵ représente le mot vide) et $\delta(q, uv) = \delta(\delta(q, u), v)$ (ici uv représente la concaténation de u et v).
Quand un automate lit un mot w, il arrive donc dans l'état $\delta(0, w)$. Si $\delta(0, w) \in \mathcal{F}$ le mot est alors accepté par l'automate sinon il est refusé.

Générateur de nombres pseudo-aléatoires

Étant donné u_0 on définit la récurrence suivante :

$$v(0) = u_0$$
$$\forall t \in \mathbb{N},\ v(t+1) = 101833 \times v(t) \mod 1\,000\,000\,007$$

Et ensuite on définit u de la façon suivante :

$$\forall t \in \mathbb{N},\ u(t) = v(t \mod 1\,000\,003)$$

L'entier u_0 vous est donné, et doit être recopié sur votre fiche réponse avec vos résultats. Une fiche réponse type vous est donnée en exemple, et contient tous les résultats attendus pour une valeur de u_0 différente de la vôtre (notée $\widetilde{u_0}$). Il vous est conseillé de tester vos algorithmes avec cet $\widetilde{u_0}$. Pour chaque calcul demandé, avec le bon choix d'algorithme le calcul ne devrait demander qu'au plus quelques secondes, jamais plus d'une minute.

Question 1 *Calculer les valeurs suivantes :*

a) $u(1) \mod 1\,000$, ***b)*** $u(42) \mod 1\,000$, ***c)*** $u(10^9) \mod 1\,000$.

Génération d'automates

On définit l'automate fini déterministe $\mathcal{E}(L, Q)$ par le quadruplet $(L, Q, \delta, \mathcal{F})$ avec $\delta(q, c) = (u(q \times L + c) \mod Q)$ et l'on a $i \in \mathcal{F}$ si et seulement si $u(Q \times L + i)$ est impair.

Question 2 *On note* TransitionsAcceptantes($\mathcal{A}$) *le nombre de paires* (q, c) *où* q *est un état et* c *une lettre, telles que* $\delta(q, c) \in \mathcal{F}$. *Calculer* TransitionsAcceptantes($\mathcal{A}$) *pour les automates* $\mathcal{E}(L, Q)$ *avec les valeurs de* L *et* Q *suivantes :*

a) $L = 2$, $Q = 100$ ***b)*** $L = 10$, $Q = 10\,000$, ***c)*** $L = 3$, $Q = 30\,000$.

Génération de mots

On définit le mot $\mathcal{M}(L, T)$ de longueur T sur l'alphabet avec L lettres comme le mot $w_0 \dots w_{T-1}$ avec, pour $0 \leq i < T$, $w_i = (u(i) \mod L)$.

Question 3 *Calculer les mots suivants :*

a) $\mathcal{M}(4, 3)$ ***b)*** $\mathcal{M}(8, 4)$ ***c)*** $\mathcal{M}(10, 5)$

On note Etat($\mathcal{A}, w$) l'état dans lequel se trouve l'automate après avoir lu le mot w, c'est à dire Etat($\mathcal{A}, w$) $= \delta(0, w)$.

Question 4 *Déterminer* Etat($\mathcal{E}(L, Q), \mathcal{M}(L, T)$) *pour les valeurs de* L, Q *et* T *suivantes :*

a) $L = 2$, $Q = 100$, $T = 100$,

b) $L = 10$, $Q = 10\,000$, $T = 50\,000$,

c) $L = 3$, $Q = 30\,000$, *et* $T = 100\,000$.

Question à développer pendant l'oral 1 *Décrire la structure de donnée utilisée pour représenter les automates et la complexité de l'algorithme pour calculer* Etat($\mathcal{E}(L, Q), \mathcal{M}(L, T)$) *en fonction de* L, Q *et* T. *Quel algorithme proposez-vous pour savoir si* $\mathcal{M}(L, T)$ *est reconnu par* $\mathcal{E}(L, Q)$, *avec quelle complexité ?*
Votre calcul de complexité ne doit pas prendre en compte le temps de construire $\mathcal{E}(L, Q)$ *et* $\mathcal{M}(L, T)$ *mais seulement le temps de calculer* Etat($\mathcal{E}(L, Q), \mathcal{M}(L, T)$) *une fois que ceux-ci sont construits.*

Accessibilité dans l'automate

Un état q de l'automate est dit accessible s'il existe un mot w tel que $q = \delta(0, w)$.

Question à développer pendant l'oral 2 *Pour un automate* $\mathcal{A}$, *on note* GrapheAccessibilité($\mathcal{A}$) *le graphe dont les nœuds sont les états de* $\mathcal{A}$ *et il y a un arc de* q *vers* q' *quand il existe une lettre* c *telle que* $\delta(q, c) = q'$. *Justifier qu'un état* q *est accessible dans l'automate* $\mathcal{A}$ *si et seulement si il existe un chemin de l'état initial de* $\mathcal{A}$ *à* q *dans le graphe* GrapheAccessibilité($\mathcal{A}$).

Question 5 *Notons* NbEtatsAccessibles($\mathcal{A}$) *le nombre d'états accessibles dans l'automate* $\mathcal{A}$. *Calculer* NbEtatsAccessibles($\mathcal{E}(L, Q)$) *pour les valeurs de* L *et* Q *suivantes :*

a) $L = 2$, $Q = 100$, ***b)*** $L = 10$, $Q = 10\,000$, ***c)*** $L = 3$, $Q = 30\,000$.

Question à développer pendant l'oral 3 *Présenter votre algorithme et sa complexité en fonction de* L *et* Q.

2 Compter les facteurs d'un mot acceptés par un automate

Étant donné $\mathcal{A}$ un automate et w un mot sur le même langage, on définit FacteursAcceptés($\mathcal{A}, w$) comme le nombre de facteurs non-vides de w acceptés par $\mathcal{A}$. Noter que l'on peut compter plusieurs fois un même mot qui apparaît plusieurs fois : un automate qui accepterait uniquement le mot "0" accepterait deux facteurs de "010" (parce que le facteur "0" y apparaît deux fois). Formellement, FacteursAcceptés($\mathcal{A}, w$) correspond au nombre de paires (i, j) avec $0 \leq i < j \leq |w|$ telles que $w[i : j]$ est un mot accepté par $\mathcal{A}$.
Dans cette section nous allons étudier deux algorithmes pour compter le nombre de facteurs d'un mot acceptés par un automate. Dans un premier temps, les automates considérés auront un grand nombre d'états tandis que dans un second temps nous regarderons le cas d'automates très petits.

2.1 Cas d'un automate avec beaucoup d'états

Question 6 *Calculer le nombre de préfixes non-vides d'un mot* $\mathcal{M}(L, T)$ *qui sont acceptés par un automate* $\mathcal{E}(L, Q)$ *pour les valeurs de* L, Q *et* T *suivantes :*

a) $L = 2$, $Q = 100$, $T = 100$

b) $L = 10$, $Q = 10\,000$, $T = 50\,000$

c) $L = 3$, $Q = 30\,000$, $T = 100\,000$.

Question 7 *En s'inspirant de l'algorithme développé à la question précédente, calculer* FacteursAcceptés($\mathcal{E}(L, Q), \mathcal{M}(L, T)$) *pour les valeurs de* L, Q *et* T *suivantes :*

a) $L = 2, Q = 100, T = 100$,

b) $L = 10, Q = 10\,000, T = 500$,

c) $L = 3, Q = 30\,000$, *et* $T = 3\,000$.

Question à développer pendant l'oral 4 *Présenter votre algorithme et sa complexité en fonction de* L, Q *et* T.

2.2 Cas d'un automate avec très peu d'états

Étant donné un automate $\mathcal{A} = (L, Q, \delta, \mathcal{F})$ et un mot $w = w_0 \ldots, w_{T-1}$, on définit $\mathcal{G}(\mathcal{A}, w)$ comme le graphe orienté dont les nœuds sont $\bot$, $\top$ ainsi que les paires (q, i) avec $0 \leq q < Q$ et $0 < i \leq T$. Le graphe contient les arcs suivants :

— $(q, i) \to (q', i+1)$ pour $0 \leq q < Q$, $q' = \delta(q, w_i)$, et $0 < i < T$,
— $\top \to (\delta(0, w_i), i+1)$ pour chaque $0 \leq i < T$,
— $(q, i) \to \bot$ pour chaque q état final de $\mathcal{A}$ et $0 < i \leq T$.

Question à développer pendant l'oral 5 *Montrer que* FacteursAcceptés$(\mathcal{A}, w)$ *est égal au nombre de chemins différents de* $\top$ *à* $\bot$ *dans* $\mathcal{G}(\mathcal{A}, w)$. *Montrer que le nombre de chemins de* $\top$ *à* $(q, i+1)$ *dépend uniquement de* δ, w_i *et des nombres de chemins de* $\top$ *à* (q', i) *pour* $0 \leq q' < Q$.

Question 8 *Déterminer* FacteursAcceptés$(\mathcal{E}(L, Q), \mathcal{M}(L, T))$ *pour les valeurs de* L, Q *et* T *suivantes :*

a) $L = 2$, $Q = 10$, $T = 1\,000$

b) $L = 10$, $Q = 30$, $T = 5\,000$

c) $L = 3$, $Q = 17$, *et* $T = 50\,000$.

Question à développer pendant l'oral 6 *Présenter votre algorithme et sa complexité en fonction de* L, Q *et* T.

3 Compter avec mises à jour

Dans cette section, on cherche un algorithme capable de résoudre le problème suivant : le mot qui nous intéresse est régulièrement modifié et l'on veut savoir rapidement après chaque modification, si le mot est accepté ou le nombre de ses facteurs qui sont acceptés par un petit automate.
Une "mise à jour" d'un mot est une paire (p, c) indiquant qu'il faut remplacer la lettre à la position p par c. Ainsi après avoir appliqué la mise à jour (p, c) le mot $w_0 \dots w_{n-1}$ devient $w_0 \dots w_{p-1}\ c\ w_{p+1} \dots w_{n-1}$.
On définit $\mathcal{L}(i, L, T)$, la i-ème mise à jour sur un mot de taille T avec un alphabet de L lettres de la façon suivante. La position de la mise à jour $\mathcal{L}(i, L, T)$ est $u(T \times L + 2i + 1) \mod T$ tandis que la nouvelle lettre est $u(T \times L + 2i) \mod L$.
Étant donné un mot initial w (sur un alphabet à L lettres et de longueur T), l'application des mises à jour de $\mathcal{L}(0, L, T)$ puis $\mathcal{L}(1, L, T)$, etc. produit une séquence de mots. On note $\mathcal{U}(i, L, T)$ le résultat des i premières mises à jour sur $\mathcal{M}(L, T)$. On a donc $\mathcal{U}(0, L, T) = \mathcal{M}(L, T)$ puis $\mathcal{U}(i+1, L, T)$ qui est le résultat de l'application de $\mathcal{L}(i, L, T)$ sur $\mathcal{U}(i, L, T)$.

Cas avec peu de mises à jour

Question 9 *En utilisant les réponses aux questions précédentes, calculer* $\sum_{0 \leq i < N}$ FacteursAcceptés$(\mathcal{E}(L, Q), \mathcal{U}(i, L, T))$ *pour les valeurs de* L, Q, T *et* N *suivantes :*

a) $L = 3$, $Q = 100$, $T = 100$, $N = 100$

b) $L = 5$, $Q = 30\,000$, $T = 100$, $N = 1\,000$

c) $L = 7$, $Q = 10$, $T = 3\,000$, $N = 100$

Question à développer pendant l'oral 7 *Expliquer l'algorithme utilisé et donner sa complexité en fonction de* Q, L, T, *et* N.

Cas d'un petit automate

Étant donnés un automate $\mathcal{A}$ et un mot w, l'effet de w sur $\mathcal{A}$, noté $\textsc{Effet}(w, \mathcal{A})$, est une fonction de l'ensemble des états de $\mathcal{A}$ vers l'ensemble des états de $\mathcal{A}$ telle que chaque état q est envoyé sur q' si q' est l'état dans lequel $\mathcal{A}$ arrive en lisant w depuis l'état q.
Étant donnés un automate $\mathcal{A}$ et un mot w non vide, $\textsc{Arb}(\mathcal{A}, w)$ est un arbre binaire. Dans cet arbre, chaque nœud correspond à un facteur v du mot w et stocke l'effet de v sur $\mathcal{A}$. En particulier la racine de $\textsc{Arb}(\mathcal{A}, w)$ correspond à w et stocke l'effet de w sur $\mathcal{A}$. $\textsc{Arb}(\mathcal{A}, w)$ peut être construit récursivement de la façon suivante :

- si $|w| = 1$ alors $\textsc{Arb}(\mathcal{A}, w)$ est un nœud sans fils,
- sinon on calcule w_1 et w_2 avec $w = w_1 w_2$ et $|w_1| \leq |w_2| \leq |w_1| + 1$. $\textsc{Arb}(\mathcal{A}, w)$ est alors un nœud qui a pour fils gauche $\textsc{Arb}(\mathcal{A}, w_1)$ et pour fils droit $\textsc{Arb}(\mathcal{A}, w_2)$.

Dans le calcul de l'arbre, pour chaque nœud $\textsc{Arb}(\mathcal{A}, u)$, on stocke $\textsc{Effet}(u, \mathcal{A})$. On note $\textsc{Accepte}(\mathcal{A}, w)$ l'entier qui vaut 1 si $\mathcal{A}$ accepte w et 0 sinon.

Question 10 *Écrire un algorithme capable de calculer* $\textsc{Arb}(\mathcal{A}, w)$ *et capable de calculer efficacement* $\textsc{Arb}(\mathcal{A}, w')$ *à partir de* $\textsc{Arb}(\mathcal{A}, w)$ *où* w' *est une mise à jour de* w*. L'utiliser pour calculer* $\sum_{0 \leq i < N} \textsc{Accepte}(\mathcal{E}(L, Q), \mathcal{U}(i, L, T))$ *pour* $T = 50\,000$*,* $N = 25\,000$*, et les valeurs de* L *et* Q *suivantes :*

a) $L = 10$, $Q = 30$ ***b)*** $L = 12$, $Q = 24$ ***c)*** $L = 5$, $Q = 25$

Question à développer pendant l'oral 8 *Expliquer l'algorithme utilisé et donner sa complexité en fonction de* Q*,* L*,* T*, et* N*.*

Question à développer pendant l'oral 9 *Expliquer quelles informations (en plus de l'effet) doivent être stockées dans chaque nœud de* $\textsc{Arb}(\mathcal{A}, w)$ *pour pouvoir répondre à* $\textsc{FacteursAcceptés}(\mathcal{A}, w)$ *et supporter des mises à jour efficaces.*

Question 11 *En déduire un algorithme capable de calculer efficacement* $\left(\sum_{0 \leq i < N} \textsc{FacteursAcceptés}(\mathcal{E}(L, Q), \mathcal{U}(i, L, T))\right) \mod 1\,000\,000$ *pour* $T = 5\,000$*,* $N = 20\,000$*, et les valeurs de* L *et* Q *suivantes :*

a) $L = 10$, $Q = 30$ ***b)*** $L = 12$, $Q = 24$ ***c)*** $L = 5$, $Q = 25$

Fiche réponse type : Comptage de mots reconnus par un automate.

$\widetilde{u_0}$: 42

Question 1

a) 986

b) 700

c) 631

Question 2

a) 113

b) 51133

c) 45445

Question 3

a) 2,2,1

b) 2,2,5,6

c) 2,6,3,0,8

Question 4

a) 12

b) 2222

c) 24124

Question 5

a) 80

b) 9999

c) 28250

Question 6

a) 56

b) 25692

c) 50493

Question 7

a) 2835

b) 65060

c) 2269352

Question 8

a) 448058

b) 5001486

c) 812170725

Question 9

a) 195813

b) 2483989

c) 246686965

Question 10

a) 18508

b) 9459

c) 12542

Question 11

a) 627390

b) 934507

c) 889835

Fiche réponse : Comptage de mots reconnus par un automate.

Nom, prénom, u_0 : .....

Question 1

a)

b)

c)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

Question 5

a)

b)

c)

Question 6

a)

b)

c)

Question 7

a)

b)

c)

Question 8

a)

b)

c)

Question 9

a)

b)

c)

Question 10

a)

b)

c)

Question 11

a)

b)

c)