Frogger

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles Normales Supérieures

Durée de l'épreuve : 4 heures

Juillet 2004

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0 clairement encadré au début de votre copie.

Important. Lorsque la description d'un algorithme est demandée, celle-ci doit être courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures sur votre copie !

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 à la main***.

Ce sujet aborde dans un cadre "ludique", les problèmes algorithmiques liés aux graphes dynamiques, i.e., aux graphes dont les arêtes changent avec le temps.

Frogger. Nous étudions ici une variante du jeu "Frogger". Il s'agit d'aider une grenouille à traverser, de bas en haut, un terrain de jeu le plus rapidement possible, sans qu'elle tombe dans l'eau. Le terrain de jeu est un cylindre recouvert d'une alternance de terre-pleins et de cours d'eau (v. FIG. 1). Sur l'eau dérivent des ponts flottants que doit emprunter la grenouille pour rejoindre le terre-plein suivant. Le terrain de jeu de largeur n et à h cours d'eau, est représenté par les cases de $\mathbb{Z}/n\mathbb{Z} \times \{0, \ldots, 2h\}$ (c'est un cylindre de périmètre n et de hauteur $2h+1$). Les cases de hauteur paire $(i, 2j)$, pour $0 \leqslant i < n$ et $0 \leqslant j \leqslant h$, sont de la terre ferme, et les cases de hauteur impaire $(i, 2j+1)$ sont des cours d'eau (illustration FIG. 1). Notez que l'indice i est pris dans $\mathbb{Z}/n\mathbb{Z}$ (i.e., que le terrain est un cylindre), et donc que les cases $(0, j)$ sont les voisines de droite des cases $(n-1, j)$.

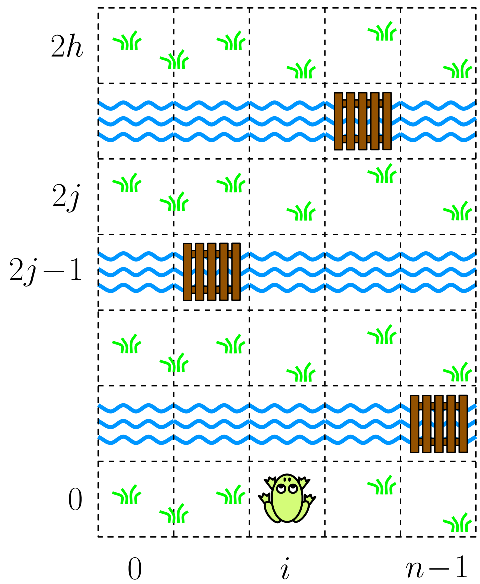

FIG. 1 – Exemple de terrain de jeu.

Le j-ième cours d'eau désigne le cours d'eau d'ordonnée $2j-1$. Le j-ième terre-plein désigne le terre-plein d'ordonnée $2j$ (les terre-pleins d'ordonnées 0 et $2h$ sont respectivement appelés les *terre-pleins de départ* et *d'arrivée*).
Chaque cours d'eau s'écoule dans un sens déterminé au départ (droite ou gauche). On attribue à chaque cours d'eau, q ponts (qui peuvent être superposés, auquel cas ils ne comptent que pour un). Chaque pont dérive d'*une case par unité de temps*, dans le sens du cours d'eau.
Pour que la grenouille puisse traverser un cours d'eau au temps t, il faut qu'elle soit à $t-1$ en face de la position où sera le pont au temps t, elle peut alors sauter sur le pont au temps t, et arriver sur l'autre berge au temps $t+1$, *en face de la position du pont au temps t* (v. FIG. 2). Remarquez que la grenouille peut rester sur le pont et dériver avec lui un certain temps, si elle le souhaite.

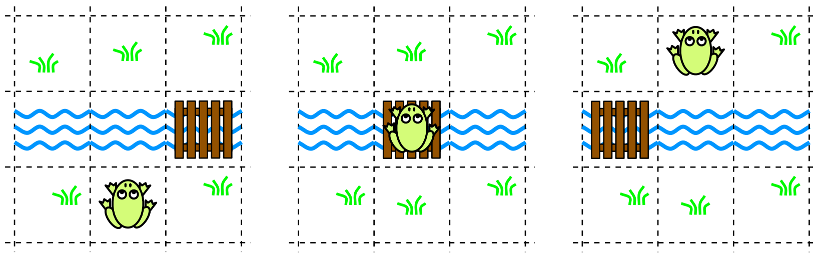

FIG. 2 – Traversée d'un cours d'eau.

Note. Toutes les évaluations de complexité des algorithmes qui vous seront demandées, doivent être exprimées en fonction des dimensions n et h du terrain de jeu, et du nombre de ponts par cours d'eau q.
Les parties 3 et 4 sont totalement indépendantes.

1 Préliminaires

On utilisera la suite (pseudo-aléatoire) d'entiers positifs $(u_n)_{0 \leqslant n \leqslant 1\,000\,000}$ définie ci-dessous :
- u_0 est donné sur votre table.
- $u_n = (16\,033 \times u_{n-1}) \mod 65\,519$, pour $n > 0$.

Question 1 *Que valent u_{787} et $u_{997\,362}$?*

Question 2 *Combien d'indices $1\,000 \leqslant i \leqslant 9\,999$ vérifient*

$$(u_i \bmod 11) \geqslant \frac{(u_{i-1} \bmod 11)}{1 + (u_{i+1} \bmod 11)} \text{ ?}$$

Idem pour $1\,000 \leqslant i \leqslant 999\,999$.

2 Le terrain de jeu

Le terrain de jeu. Le terrain de jeu T est défini par le quadruplet $\langle n, h, q, S \rangle$ avec $n, h, q \in \mathbb{N}^*$ et $S \in \{\ominus, \leftleftarrows, \rightleftarrows\}$. Les cases du terrain de jeu sont indexées par $(i, j) \in \mathbb{Z}/n\mathbb{Z} \times \{0, 1, \ldots, 2h\}$. On attribue à chaque cours d'eau q ponts flottants. Initialement, à $t = 0$, les coordonnées du k-ième pont du j-ième cours d'eau sont :

$$\left(\left(u_{(q \cdot j + k - 1)} \bmod n \right), \ 2j - 1 \right) \quad \text{pour } 1 \leqslant k \leqslant q$$

Plusieurs ponts peuvent avoir les mêmes coordonnées initiales, dans ce cas ils ne font qu'un. Si $S = \ominus$, tous les ponts sont immobiles. Si $S = \leftleftarrows$, tous les ponts se déplacent d'une case vers la gauche par unité de temps. Si $S = \rightleftarrows$, les ponts du j-ième cours d'eau se déplacent d'une case par unité de temps vers la gauche si j est impair, et vers la droite si j est pair.

Le but du jeu. À $t = 0$, la grenouille est sur la case $(0, 0)$. Le but du jeu est de guider la grenouille pour atteindre au plus tôt le terre-plein d'arrivée ; la *date d'arrivée optimale* est le plus petit instant t auquel la grenouille peut atteindre le terre-plein d'arrivée.

Question 3 *Combien il y a-t-il de ponts (distincts) dans le 5ème cours d'eau ? au total ? dans les terrains de jeu suivants :*

$$\langle n = 10, h = 10, q = 5, \ominus\rangle, \quad \langle n = 20, h = 20, q = 10, \ominus\rangle,$$
$$\langle n = 200, h = 100, q = 18, \ominus\rangle, \quad \langle n = 200, h = 500, q = 50, \ominus\rangle \text{ ?}$$

Expliquez comment vous avez procédé.

Question 4 *Donnez la liste ordonnée des distances entre les ponts du 5ème cours d'eau (i.e., la liste des nombres de cases vides entre deux ponts distincts consécutifs), dans les terrains de jeu :*

$$\langle n = 10, h = 10, q = 5, \ominus\rangle \quad \textit{et} \quad \langle n = 200, h = 100, q = 18, \ominus\rangle.$$

(N'oubliez pas que le terrain de jeu est cylindrique !)

Question 5 *Donnez l'expression de la position au temps $t \in \mathbb{N}$ du k-ième pont (pas nécessairement distinct des autres) du j-ième cours d'eau dans le terrain $T = \langle n, h, q, S\rangle$ en fonction des paramètres.*
Donnez les positions à l'instant $t = 311$, du 9ème pont du 17ème et du 18ème cours d'eau du terrain $\langle n = 200, h = 500, q = 15, \rightleftarrows\rangle$.

3 Grenouille courtes-pattes

Grenouille courtes pattes. Dans cette partie, la grenouille ne peut se déplacer que d'*une case par unité de temps*, ou bien rester immobile.

Pour les deux questions suivantes, on se limite aux terrains de jeu avec *un seul pont* par cours d'eau, et où tous les ponts sont *immobiles*.

Question 6 *Proposez un algorithme qui calcule la date d'arrivée optimale de la grenouille sur un terrain de type $\langle n, h, \mathbf{q = 1}, \boldsymbol{\ominus}\rangle$.*
Donnez sa complexité en temps.

Question 7 *Quelle est la date d'arrivée optimale et la position de la grenouille à l'arrivée pour chacun des terrains suivants :*

$$\langle n = 20, h = 20, q = 1, \ominus\rangle, \quad \langle n = 50, h = 20, q = 1, \ominus\rangle,$$
$$\langle n = 500, h = 10\,000, q = 1, \ominus\rangle, \quad \langle n = 1\,000, h = 500\,000, q = 1, \ominus\rangle \text{ ?}$$

On considère à présent des terrains ayant q *ponts* par cours d'eau, qui se déplacent *tous d'une case vers la gauche par unité de temps.*

Question 8 *Démontrez que, dans toute solution optimale, la grenouille ne se déplace que dans certaines directions.*
Proposez un algorithme glouton qui calcule la date d'arrivée optimale dans un terrain de type $\langle n, h, q, \boldsymbol{\leftleftarrows} \rangle$.
Donnez sa complexité en temps.

Question 9 *Quelle est la date d'arrivée optimale et la position de la grenouille à l'arrivée pour chacun des terrains suivants :*

$$\begin{aligned} \langle n = 10, h = 10, q = 2, \leftleftarrows \rangle, &\quad \langle n = 20, h = 20, q = 3, \leftleftarrows \rangle, \\ \langle n = 100, h = 1\,000, q = 4, \leftleftarrows \rangle, &\quad \langle n = 10\,000, h = 3\,000, q = 300, \leftleftarrows \rangle. \\ \textit{et} &\quad \langle n = 100\,000, h = 900, q = 1\,000, \leftleftarrows \rangle. \end{aligned}$$

On considère à présent des terrains de type $\rightleftarrows$ ayant q *ponts* par cours d'eau.

Question 10 *Démontrez qu'il existe une solution optimale, où la grenouille ne se déplace jamais sur un terre-plein dans le sens du cours d'eau suivant.*

Question 11 *Proposez un algorithme qui calcule la date d'arrivée optimale dans un terrain de type $\langle n, h, q, \boldsymbol{\rightleftarrows} \rangle$.*
Donnez sa complexité en temps.

Question 12 *Quelle est la date d'arrivée optimale et la position de la grenouille à l'arrivée pour chacun des terrains suivants :*

$$\begin{aligned} \langle n = 11, h = 11, q = 2, \rightleftarrows \rangle, &\quad \langle n = 21, h = 21, q = 3, \rightleftarrows \rangle, \\ \langle n = 101, h = 1\,001, q = 4, \rightleftarrows \rangle, &\quad \langle n = 10\,001, h = 3\,001, q = 301, \rightleftarrows \rangle. \\ \textit{et} &\quad \langle n = 100\,001, h = 901, q = 1\,001, \rightleftarrows \rangle. \end{aligned}$$

4 Grenouille grandes pattes

Grenouille grandes pattes. Dans cette section, la grenouille peut se déplacer d'*une ou deux cases par unité de temps vers la droite ou vers la gauche*, et toujours que d'*une case vers le haut ou vers le bas*, ou rester immobile.

Question 13 *Proposez un algorithme qui calcule, pour chaque case de terre-plein, la date à laquelle la grenouille peut l'atteindre au plus tôt. Décrivez-le soigneusement. Prouvez que votre algorithme est correct, i.e., qu'il calcule correctement cette date optimale.*
Donnez la complexité en temps et en mémoire de votre algorithme.

Question 14 *Donnez la date minimale d'atteinte des cases suivantes des terrains suivants :*

$$\begin{aligned} \textit{la case } (5, 20) \textit{ du terrain } &\langle n = 10, h = 10, q = 2, \rightleftarrows \rangle, \\ \textit{la case } (15, 30) \textit{ du terrain } &\langle n = 20, h = 20, q = 3, \rightleftarrows \rangle, \\ \textit{la case } (15, 90) \textit{ du terrain } &\langle n = 50, h = 50, q = 6, \rightleftarrows \rangle, \\ \textit{la case } (15, 190) \textit{ du terrain } &\langle n = 100, h = 100, q = 7, \rightleftarrows \rangle, \\ \textit{et la case } (15, 990) \textit{ du terrain } &\langle n = 500, h = 500, q = 10, \rightleftarrows \rangle. \end{aligned}$$

Donnez pour chacun de ces terrains les dates d'arrivée optimales.

On ajoute sur chaque terre-plein r murs. Le k-ième mur du j-ième terre-plein a pour coordonnée :

$$(\ (u_{(q \cdot h + r \cdot j + k - 1)} \mod n) \ , \ 2j \ ) \quad \text{pour } 1 \leqslant j \leqslant h-1 \text{ et } 1 \leqslant k \leqslant r.$$

Les terre-pleins de départ et de d'arrivée n'ont pas de murs. Comme précédemment si plusieurs murs ont les mêmes coordonnées, ils ne font qu'un. La grenouille ne peut ni franchir, ni arriver sur un mur.

Question 15 *Exhibez un exemple de terrain de type* $\langle n, h, \mathbf{q = 1}, \boldsymbol{\leftleftarrows}, \mathbf{r = 1} \rangle$, *(où vous placerez où vous le souhaiterez les murs et les ponts, i.e., indépendamment des valeurs de la suite* (u_n)*), où la grenouille reste sur un pont flottant pendant un plusieurs unités de temps dans toute solution optimale.*
Exhibez un exemple de terrain de type $\langle n, h, \mathbf{q = 3}, \boldsymbol{\leftleftarrows}, \mathbf{r = 1} \rangle$, *où la grenouille doit revenir sur un terre-plein de niveau inférieur dans toute solution optimale.*

Question 16 *Proposez un algorithme qui calcule la date minimale d'atteinte de toutes les cases des terre-pleins d'un terrain quelconque* $\langle n, h, q, S, r \rangle$. *Justifiez que votre algorithme calcule bien la date optimale.*
Donnez-en la complexité en temps et en mémoire.

Question 17 *Donnez la date minimale d'atteinte de la première case qui n'est pas un mur, à droite des cases spécifiées des terrains suivants :*

$$\begin{aligned}
\textit{la case } (5, 20) \textit{ du terrain } &\langle n = 10, h = 10, q = 2, \rightleftarrows, r = 1 \rangle, \\
\textit{la case } (15, 30) \textit{ du terrain } &\langle n = 20, h = 20, q = 3, \rightleftarrows, r = 2 \rangle, \\
\textit{la case } (15, 90) \textit{ du terrain } &\langle n = 50, h = 50, q = 6, \rightleftarrows, r = 5 \rangle, \\
\textit{la case } (15, 190) \textit{ du terrain } &\langle n = 100, h = 100, q = 7, \rightleftarrows, r = 10 \rangle, \\
\textit{et la case } (15, 590) \textit{ du terrain } &\langle n = 300, h = 300, q = 10, \rightleftarrows, r = 15 \rangle.
\end{aligned}$$

Donnez pour chacun de ces terrains les dates d'arrivée optimales.

Un trésor est déposé sur chaque terre-plein. La position du trésor du j-ième terre-plein est :

$$(\ (u_{((q+r) \cdot h + j)} \mod n) \ , \ 2j \ ) \quad \text{pour } 1 \leqslant j \leqslant h$$

Le terre-plein de départ n'a pas de trésor. Si un trésor a la même coordonnée qu'un mur, le mur est effacé par le trésor. On note $\langle n, h, q, S, r, \mathbb{T} \rangle$ ce nouveau type de terrain. La grenouille doit ramasser ces trésors *dans l'ordre, de bas en haut*. Le but est de minimiser la date où la grenouille ramasse le dernier trésor, sur le terre-plein d'arrivée.

Question 18 *Exhibez un exemple où la grenouille revient sur un terre-plein inférieur dans une solution optimale.*
Proposez un algorithme qui calcule la date d'arrivée optimale. Justifiez qu'il calcule bien la date optimale et que la grenouille aura bien ramassé tous les trésors.
Donnez-en la complexité en temps et en mémoire.

Question 19 *Donnez le temps minimal que mettra la grenouille pour ramasser le cinquième trésor, puis pour ramasser tous les trésors, dans les terrains suivants :*

$$\begin{aligned}
\langle n=10, h=10, q=2, \leftleftarrows, r=1, \mathbb{T}\rangle, &\quad \langle n=20, h=20, q=3, \rightleftarrows, r=2, \mathbb{T}\rangle, \\
\langle n=50, h=50, q=6, \rightleftarrows, r=5, \mathbb{T}\rangle, &\quad \langle n=100, h=100, q=7, \rightleftarrows, r=10, \mathbb{T}\rangle, \\
\textit{et} &\quad \langle n=300, h=300, q=10, \rightleftarrows, r=15, \mathbb{T}\rangle.
\end{aligned}$$

Les trésors sont maintenant des mouches que la grenouille souhaite gober. La position initiale de chaque mouche, à $t = 0$, est celle du trésor correspondant dans les questions précédentes. Les mouches se déplacent d'une case par unité de temps : *vers la gauche pour les terre-pleins impairs, et vers la droite pour les terre-pleins pairs.* Contrairement aux trésors, les mouches n'effacent pas les murs lorsqu'elles leurs sont superposées, mais elles les survolent. De plus, lorsqu'une mouche est sur un mur, la grenouille ne peut pas la gober. Nous noterons $\langle n, h, q, S, r, \mathbb{M}\rangle$ ce nouveau type de terrain. Nous cherchons à calculer le temps minimum que met la grenouille pour gober toutes les mouches *dans l'ordre, de bas en haut.*

Question 20 *Proposez un algorithme qui calcule le temps minimum que met la grenouille pour gober toutes les mouches. Justifiez l'optimalité de la solution calculée.*
Précisez la complexité en temps et en mémoire de votre algorithme.

Question 21 *Donnez le temps minimal que mettra la grenouille pour gober la cinquième mouche, puis pour gober toutes les mouches, dans les terrains suivants :*

$$\begin{aligned}
\langle n=10, h=10, q=2, \leftleftarrows, r=1, \mathbb{M}\rangle, &\quad \langle n=20, h=20, q=3, \rightleftarrows, r=2, \mathbb{M}\rangle, \\
\langle n=50, h=50, q=6, \rightleftarrows, r=5, \mathbb{M}\rangle, &\quad \langle n=100, h=100, q=7, \rightleftarrows, r=10, \mathbb{M}\rangle, \\
\textit{et} &\quad \langle n=300, h=300, q=10, \rightleftarrows, r=15, \mathbb{M}\rangle.
\end{aligned}$$

- Fin -