Labyrinthes

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

Durée de l'épreuve: 3 heures 30 minutes

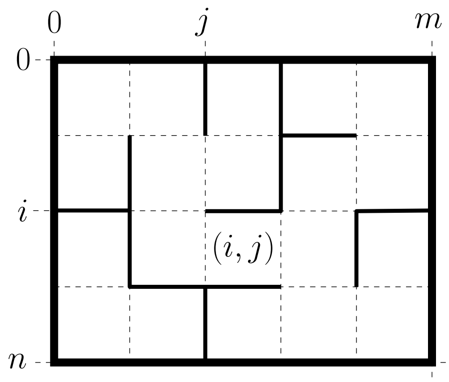
Juillet 2005

ATTENTION !

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

Important.
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***.

Labyrinthes. Nous considérons un rectangle de $n \times m$ *cases* dans $\mathbb{Z}^2$ (dont les deux axes sont respectivement orientés vers le bas et vers la droite – comme pour les matrices). Chaque case est numérotée (i, j) par les coordonnées de son coin supérieur gauche, $0 \leqslant i < n$ et $0 \leqslant j < m$. Un *aménagement* de ce rectangle est un ensemble de murs (horizontaux et verticaux) tel que tous les murs extérieurs sont présents. Nous dirons qu'un aménagement est un *labyrinthe* si tous les coins de toutes les cases sont l'origine d'au moins un mur (c'est-à-dire, s'il n'existe que des «couloirs», et pas de «pièce»).
La figure suivante donne un exemple de labyrinthe $n \times m$ (remarquez qu'un labyrinthe n'est pas nécessairement d'un seul morceau et que les cases en bas à gauche sont inaccessibles depuis les cases en haut à droite).



Un aménagement `lab` sera codé dans votre programme par quatre variables : ses deux dimensions `labN` et `labM`, et deux tableaux *unidimensionnels* d'entiers `labH` et `labV`, de longueur `labN`·`labM`, décrivant respectivement les murs horizontaux et verticaux internes :

- $\texttt{labH}[i \cdot \texttt{labM} + j] = \begin{cases} 1 & \text{si le mur } (i,j)\text{—}(i,j+1) \text{ existe} \\ 0 & \text{sinon} \end{cases} \quad \Big(\text{avec } 1 \leqslant i < \texttt{labN} \text{ et } 0 \leqslant j < \texttt{labM}.\Big)$
- $\texttt{labV}[i \cdot \texttt{labM} + j] = \begin{cases} 1 & \text{si le mur } \begin{array}{c}(i,j)\\ | \\ (i+1,j)\end{array} \text{ existe} \\ 0 & \text{sinon} \end{cases} \quad \Big(\text{avec } 0 \leqslant i < \texttt{labN} \text{ et } 1 \leqslant j < \texttt{labM}.\Big)$

Les autres entrées des tableaux ne sont pas utilisées.
Par exemple, le labyrinthe `fig` ci-dessus est codé par les quatre variables `figN`, `figM`, `figH` et `figV`, où `figN = 4`, `figM = 5`, le tableau des murs horizontaux vaut :

`figH = [ 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1 0 0 ]`

et le tableau des murs verticaux vaut :

`figV = [ 0 0 1 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 ]`

Note. Toutes les évaluations de complexité des algorithmes qui vous seront demandées à l'oral, doivent être exprimées en fonction des dimensions n et m du labyrinthe.

Les sections 2, 4, 5 et 6 sont très largement indépendantes les unes des autres et peuvent être résolues dans un ordre arbitraire sans avoir répondu aux précédentes.

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.
– $u_n = (16007 \times u_{n-1}) \bmod 65521$, pour $n > 0$.

Question 1 *Que valent **a)** u_{789} et **b)** u_{992362} ?*

Question 2 ***a)** Combien d'indices $1000 \leqslant i \leqslant 9\,999$ vérifient*

$$\frac{1 + (u_{i-1} \bmod 5)}{3} \leqslant (u_i \bmod 6) \;?$$

***b)** Idem pour $1000 \leqslant i \leqslant 999\,999$.*

2 Génération pseudo-aléatoire de labyrinthes

Pour $p \in \{0, \ldots, 1000\}$ et $n, m \in \mathbb{N}^*$, nous définissons l'aménagement `lab_`n`x`m`_`p de taille $n \times m$ de la façon suivante :

$$\texttt{lab\_}n\texttt{x}m\texttt{\_}p\texttt{H}[z] = \begin{cases} 1 & \text{si } (u_z \bmod 1000) < p \\ 0 & \text{sinon.} \end{cases}$$

$$\text{et} \quad \texttt{lab\_}n\texttt{x}m\texttt{\_}p\texttt{V}[z] = \begin{cases} 1 & \text{si } (u_{nm+z} \bmod 1000) < p \\ 0 & \text{sinon.} \end{cases}$$

Question 3 *Dessinez les murs du sous-rectangle 4×4, dont le coin supérieur gauche est $(50, 50)$ et le coin inférieur droit est $(54, 54)$, dans l'aménagement `lab_100x100_400`.*

Question 4 *Pour quels intervalles de valeurs de $p \in \{0, \ldots, 1000\}$, les aménagements suivants sont-ils des labyrinthes : **a)** `lab_10x10_`p, **b)** `lab_25x30_`p, **c)** `lab_60x50_`p, **d)** `lab_100x110_`p, **e)** `lab_210x200_`p et **f)** `lab_500x600_`p ?*

★ *Vous présenterez à l'oral l'algorithme que vous avez utilisé, ainsi que sa complexité.*

Étant donné un aménagement `lab`, nous cherchons à lui ajouter le plus petit nombre possible de murs d'un seul type, vertical ou horizontal, pour en faire un labyrinthe.

Question 5 *Quel est le nombre minimum de murs verticaux nécessaires pour compléter en labyrinthes chacun des aménagements suivants : **a)** `lab_10x10_200`, **b)** `lab_25x30_300`, **c)** `lab_60x50_300`, **d)** `lab_100x110_300`, **e)** `lab_110x100_400`, **f)** `lab_210x200_300` et **g)** `lab_500x600_350` ?*

★ *Vous présenterez à l'oral l'algorithme que vous avez utilisé, ainsi que sa complexité.*

Indication. *Peut-on traiter chaque colonne indépendamment ?*

3 Chargement des données pour les sections suivantes

Les données. ***À partir de cette section, les questions portent sur des labyrinthes donnés dans des fichiers que vous devez charger depuis le disque dur de l'ordinateur.*** Les fichiers sont stockés dans le répertoire `C:\LAB` sous Windows et dans le répertoire `/home/guest/lab/` sous Linux. Reportez-vous à la fiche à part pour la procédure (très simple) à suivre pour inclure ces données dans votre programme en fonction du langage et du système d'exploitation que vous utilisez.
Ces fichiers définissent en mémoire les deux constantes entières `lab`x`N`, `lab`x`M`, et deux tableaux constants `lab`x`H`, `lab`x`V`, correspondant à 13 aménagements `lab`x avec $x \in Z$ où

$$Z = \{10, 11, 12, 25, 26, 27, 50, 51, 53, 80, 81, 82, 100\}$$

de taille $x \times x$ (soit 26 constantes en tout, et 26 tableaux constants).

4 Connexité

On dit que deux cases sont *voisines* dans un labyrinthe si elles ont un côté commun qui n'est pas un mur. Un *chemin* d'une case (i, j) à une case (k, l) dans un labyrinthe est une suite finie de cases $(i_1, j_1), \ldots, (i_\ell, j_\ell)$, telle que $(i_1, j_1) = (i, j)$, $(i_\ell, j_\ell) = (k, l)$ et telle que les cases (i_q, j_q) et (i_{q+1}, j_{q+1}) sont voisines pour tout $1 \leqslant q < \ell$. On dit qu'un labyrinthe est *connexe* s'il existe en chemin entre toute paire de cases, c'est-à-dire si toute case est accessible par un chemin depuis toute autre case.

Question 6 *Soit X l'ensemble des $x \in Z$, tels que l'aménagement* `lab`x *(chargé dans le fichier à la section précédente) est un labyrinthe connexe. Que vaut X ?*

★ *Vous présenterez à l'oral l'algorithme que vous avez utilisé, ainsi que sa complexité.*

Indication. *On pourra chercher à visiter (récursivement ?) toutes les cases accessibles depuis un point de départ arbitraire. Un tableau de booléens indiquant si telle ou telle case a déjà été visitée pourra être utile pour éviter de revisiter des cases.*

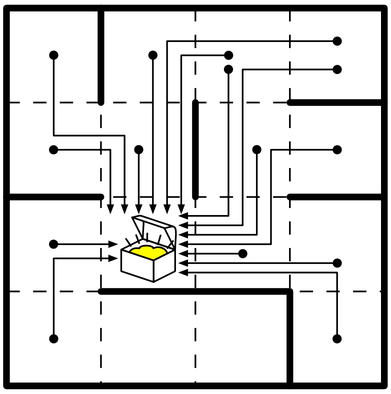
Remarque. La détermination exacte de l'ensemble X n'est pas indispensable pour la suite. Si vous n'avez pas déterminé X, vous pourrez toujours répondre aux questions en prenant $x \in Z$.

Les deux sections suivantes sont totalement indépendantes
et peuvent être traitées séparément.

5 Chasse au trésor

Thésée décide de partir à la chasse au trésor que Dédale a caché dans le labyrinthe. Thésée est équipé d'un détecteur de métaux et peut se faire déposer par hélicoptère sur n'importe quelle case pour commencer sa chasse. Une fois sur la case de départ (i, j), Thésée connaît, grâce à son détecteur, la position (k, l) du trésor et adopte la stratégie suivante pour l'atteindre : il choisit une des directions possibles qui le *rapproche strictement* du trésor au sens de la distance ℓ_1 dans $\mathbb{Z}^2$ (c'est-à-dire, indépendamment des murs du labyrinthe) ; si aucune de ces directions n'est possible, il abandonne.[1]

Étant donnée une position (i, j) du trésor, nous dirons qu'un chemin de la case (k, l) à la case (i, j) est *gagnant* pour Thésée, si chaque pas du chemin rapproche strictement du trésor au sens de la distance ℓ_1 dans $\mathbb{Z}^2$. Dans la figure suivante, exactement 15 chemins gagnants mènent au trésor.



Question 7 *Quel est le nombre de chemins n'allant que vers le Nord et vers l'Est qui atteignent la case* $(\lfloor x/2 \rfloor, \lfloor x/2 \rfloor)$ *du labyrinthe* `lab`x *avec* $x \in X$*, où* $\lfloor z \rfloor$ *désigne la partie entière inférieure de* z *?*

★ *Vous présenterez à l'oral un algorithme efficace qui calcule pour chaque case* (i, j)*, le nombre de chemins n'allant que vers le Nord et vers l'Est qui l'atteignent cette case.*

Indication. *On pourra chercher à exprimer le nombre de chemins n'allant que vers le Nord et vers l'Est atteignant une case en fonction des nombres de tels chemins atteignant ses cases voisines.*

Question 8 ★ *Vous présenterez à l'oral un algorithme efficace qui calcule pour chaque position du trésor* (i, j)*, le nombre de chemins gagnants distincts pour Thésée.*

Dédale cherche à minimiser les chances de perdre son trésor et décide donc de le placer sur une case qui minimise le nombre de chemins gagnants pour Thésée.

Question 9 *Pour chaque labyrinthe* `lab`x *avec* $x \in X$*, donnez une position du trésor qui minimise le nombre de chemins gagnants distincts. Quel est alors ce nombre minimum de chemins gagnants pour chaque* $x \in X$ *?*

Question 10 *Inversement, pour chaque labyrinthe* `lab`x *avec* $x \in X$*, donnez la liste des positions de départ pour Thésée qui maximisent le nombre de chemins possibles distincts (c'est-à-dire le nombre de chemins qu'il peut suivre depuis cette position, en visant une case donnée et en s'en rapprochant strictement à chaque pas). Quel est ce nombre maximum de chemins pour chaque* $x \in X$ *?*

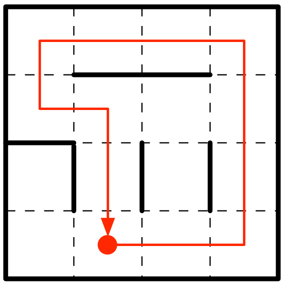
★ *Vous présenterez à l'oral l'algorithme que vous avez utilisé, ainsi que sa complexité.*

Question 11 *Pour quels labyrinthes* `lab`x *avec* $x \in X$*, Thésée a-t-il une chance de trouver le trésor ? (c'est-à-dire, existe-t-il une position de départ optimale pour Thésée qui mène avec le détecteur de métaux à une position optimale pour Dédale ?)*

[1] La distance ℓ_1 de (u, v) à (u', v') est : $|u - u'| + |v - v'|$.

6 Vidéo-surveillance

Conformément à l'article 10 de la loi n° 95.73 du 21 janvier 1995, Dédale souhaite équiper son labyrinthe d'un système de surveillance. Tout d'abord, il désire éliminer les cycles de son labyrinthe. Un *cycle* est une suite de cases voisines distinctes de longueur au moins trois, qui revient à son point de départ (voir illustration ci-dessous).



Question 12 *Notons Y l'ensemble des $x \in X$ tels que* `lab`x *n'a pas de cycle. Que vaut Y ?*

★ *Vous présenterez à l'oral l'algorithme que vous avez utilisé, ainsi que sa complexité.*

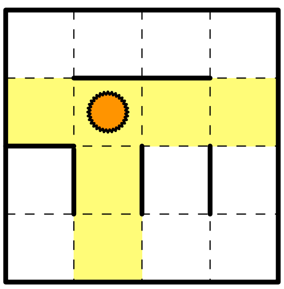
Indication. *On pourra modifier le parcours obtenu à la question 6 et vérifier qu'on ne visite pas une même case par deux chemins différents.*

Remarque. La détermination exacte de l'ensemble Y n'est pas indispensable pour la suite. Si vous n'avez pas déterminé Y, vous pourrez toujours répondre aux questions en prenant $x \in Z$ ou $x \in X$.

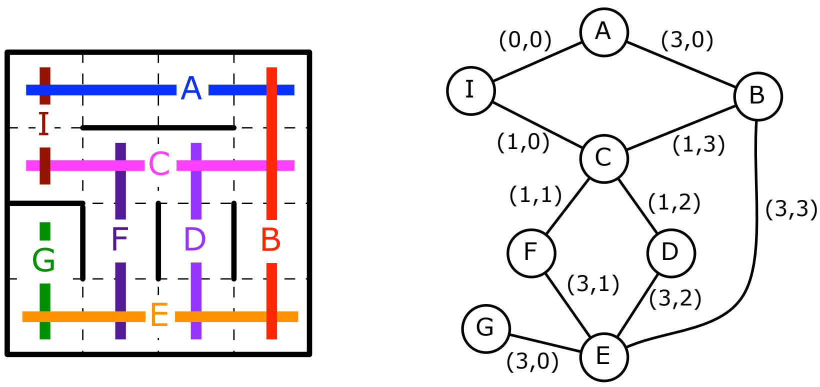
Dédale souhaite ensuite obtenir un plan synthétique de ses labyrinthes. Pour cela, il appelle *couloir* horizontal (resp. vertical) d'un labyrinthe tout ensemble maximal de plusieurs cases voisines horizontalement (resp. verticalement) qui ne sont pas séparées par un murs. Puis, il appelle *intersection* toute case qui se trouve à l'intersection de deux couloirs distincts. Par exemple, le graphe ci-dessus compte huit couloirs et dix intersections. Un exemple de couloirs horizontal et vertical sont : de la case $(3, 0)$ à la case $(3, 3)$, et de la case $(1, 2)$ à $(3, 2)$; un exemple d'intersection est la case $(3, 2)$.

Question 13 *Pour chaque $y \in Y$, déterminez le nombre de couloirs et d'intersections du labyrinthe* `lab`y*.*

Dédale désire placer des caméras de vidéo-surveillance dans son labyrinthe pour pouvoir en surveiller toutes les cases. Un tel placement sera dit *valide*. Une caméra voit dans les quatre directions (Nord, Sud, Est, Ouest), mais sa vue est arrêtée par les murs. Chaque caméra sera placée au centre de la case qui lui est affectée. Dans l'exemple ci-dessous, la caméra placée sur la case $(1, 1)$ surveille exactement 6 cases.



Pour résoudre son problème de vidéo-surveillance, il décide donc de le représenter son labyrinthe sous forme d'un graphe en représentant chaque couloir par un sommet et en reliant deux sommets entre eux si les couloirs correspondants s'intersectent. Les arêtes de ce graphe peuvent être donc étiquetées par les cases des intersections correspondantes. La figure ci-dessous illustre cette représentation.



Question 14 *Pour chaque labyrinthe* `laby`y *avec* $y \in Y$*, déterminez le nombre minimum de caméras nécessaires.*

★ *Vous présenterez à l'oral l'algorithme que vous avez utilisé, ainsi que sa complexité.*

Indication. *Remarquez qu'il est préférable de placer les caméras sur certaines cases plutôt que d'autres. Ensuite, les graphes correspondants aux labyrinthes* `laby`y*, pour* $y \in Y$*, peuvent-ils contenir des cycles ? À quoi correspond pour le graphe un placement valide des caméras vidéo ? Remarquez que tout graphe sans cycle, ayant au moins une arête, a un sommet qui a un unique voisin.*