

# Dominos

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

Durée de l'épreuve: 3 heures 30 minutes

Juillet 2011

**ATTENTION !**

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

## Important.

Sur votre table est indiqué 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  $\tilde{u}_0$  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  $\tilde{u}_0$  au lieu de  $u_0$ . Vous indiquerez vos réponses (correspondant à votre  $u_0$ ) sur la seconde et vous la remettrez à l'examineur 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 Génération de dominos

Un domino est la donnée de deux entiers compris entre 0 et  $M$ . La valeur de  $M$  sera spécifiée pour chaque question du sujet. L'orientation d'un domino au moment du tirage n'a pas d'importance :  $\langle 1 : 2 \rangle$  et  $\langle 2 : 1 \rangle$  sont deux façons de représenter le même domino et l'une ou l'autre pourra être utilisée.

Considérons la suite d'entiers  $(u_k)_k$  définie par

$$u_{k+1} = 15091 \times u_k \pmod{64007}.$$

La suite  $(u_k)_k$  va servir à sélectionner les dominos disponibles. Les points sur les deux faces des dominos sont donnés par  $u_{2i} \pmod{M+1}$  et  $u_{2i+1} \pmod{M+1}$  pour  $i = 0, 1, 2, \dots$

**Question 1** *Quels sont les deux premiers dominos pour* **a)**  $M = 6$  ? **b)**  $M = 13$  ?

Un jeu de dominos ne contient généralement pas de doublons. Il faut donc interdire d'avoir deux fois le même domino. C'est l'hypothèse qui sera faite pour la suite de ce sujet et il reste donc à éliminer les doublons du tirage.

**Question 2** *Pour*  $M = 15$ , *combien faut-il tirer de dominos pour qu'il en reste*  $N$  *une fois que les doublons ont été retirés et quel est le dernier domino tiré pour* **a)**  $N = 10$  ? **b)**  $N = 20$  ? **c)**  $N = 50$  ?

**Question à développer pendant l'oral :** Quelle est la complexité de votre algorithme ? Combien peut valoir  $N$  au maximum ?

Dans les questions suivantes, on s'assurera de précalculer les séquences  $D_{M,N}$  de  $N$  dominos différents (portant des points entre 0 et  $M$ ) de manière à pouvoir y accéder en temps constant.

# 2 Train de dominos

On considère une version simplifiée du jeu de dominos où l'objectif est simplement de faire la plus longue suite possible avec les dominos piochés. Un train de longueur  $\ell$  est une suite  $\langle g_k : d_k \rangle_{1 \leq k \leq \ell}$  de dominos orientés de telle sorte que  $d_k = g_{k+1}$ , c'est-à-dire que deux dominos voisins dans le train ont le même nombre de points sur les deux faces en contact. Chaque domino du train n'apparaît qu'une seule fois.

Ainsi, étant donnés les six dominos

$$\langle 0 : 5 \rangle \quad \langle 1 : 2 \rangle \quad \langle 2 : 3 \rangle \quad \langle 2 : 5 \rangle \quad \langle 3 : 4 \rangle \quad \langle 3 : 5 \rangle$$

on peut construire le train de longueur 4 suivant :

$$\langle 2 : 3 \rangle \langle 3 : 5 \rangle \langle 5 : 2 \rangle \langle 2 : 1 \rangle.$$

**Question 3** *Combien peut-on construire de trains*  $\langle p : q \rangle \langle q : r \rangle$  *de longueur 2 (avec*  $p < r$  *en prenant des dominos* **a)** *dans*  $D_{6,10}$  ? **b)** *dans*  $D_{24,20}$  ?

**Question 4** Quelle est la longueur du plus long train qu'il est possible de construire avec les dominos **a)** de  $D_{6,10}$  ? **b)**  $D_{12,10}$  ? **c)**  $D_{24,20}$  ? **d)**  $D_{24,30}$  ?

**Question à développer pendant l'oral :** Quelle est la complexité en temps de votre algorithme ?

### 3 Heuristique : respect de l'ordre de tirage

L'algorithme que vous avez utilisé à la question 4 est vraisemblablement trop lent pour des jeux de dominos plus gros. Pour la suite, on ne va donc plus chercher le train le plus long mais essayer d'en trouver un suffisamment long.

Pour accélérer la recherche de trains, on peut se restreindre à des trains d'une certaine forme. Dans un premier temps, on ne considère que les trains dont les dominos apparaissent dans l'ordre de leur tirage. Autrement dit, un train  $\langle g_k : d_k \rangle_{1 \leq k \leq \ell}$  n'est maintenant valable que si, pour tout  $i < j$ , le domino  $\langle g_i : d_i \rangle$  a été pioché avant le domino  $\langle g_j : d_j \rangle$ .

**Question à développer pendant l'oral :** Soit  $L_{n,v}$  la longueur du plus long train dont les dominos ont été pris dans l'ordre parmi les  $n$  premiers dominos piochés et dont l'extrémité droite porte  $v$  points. Exprimez  $L_{n,v}$  en fonction des  $L_{n-1,v'}$  pour  $v'$  quelconque.

**Question 5** Quelle est la longueur du plus long train respectant l'ordre du tirage qu'il est possible de faire avec les dominos pris **a)** dans  $D_{6,10}$  ? **b)** dans  $D_{24,30}$  ? **c)** dans  $D_{100,2000}$  ?

**Question à développer pendant l'oral :** Quelle est la complexité en temps et en espace de votre algorithme ?

Plutôt que de respecter strictement l'ordre de tirage, on peut découper l'ensemble de départ  $D_{M,N}$  en blocs de  $K$  dominos : les  $K$  premiers dominos piochés, les  $K$  suivants, etc. Dorénavant, le train doit respecter seulement l'ordre des blocs. Autrement dit, un train  $\langle g_k : d_k \rangle_{1 \leq k \leq \ell}$  devient maintenant valable si, pour tout  $i < j$ , le domino  $\langle g_i : d_i \rangle$  a été pris dans le même bloc ou dans un bloc précédent celui du domino  $\langle g_j : d_j \rangle$ .

**Question 6** Quelle est la longueur du plus long train respectant l'ordre des blocs avec  $K = 5$  qu'il est possible de faire avec les dominos pris **a)** dans  $D_{6,10}$  ? **b)** dans  $D_{24,30}$  ? **c)** dans  $D_{100,2000}$  ?

**Question à développer pendant l'oral :** Si l'objectif est de minimiser le temps d'exécution de l'algorithme, quelle valeur de  $K$  choisir en fonction de  $M$  et  $N$  ?

### 4 Heuristique : guidage par voisinage

Voyons maintenant une façon de choisir un train qui ne soit plus guidée par l'ordre de tirage. Pour chaque valeur  $v$ , on peut noter  $V_v$  les dominos piochés portant  $v$  points d'un côté. On construit alors un train en choisissant la valeur  $v$  pour laquelle  $V_v$  est non vide et contient le moins de dominos. S'il y a plusieurs valeurs de  $v$  possibles, on choisit la plus petite.

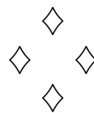
On prend comme premier domino celui de  $V_v$  qui est double (c'est-à-dire  $\langle v : v \rangle$ ) s'il existe ou celui qui porte la valeur  $v' \neq v$  pour laquelle  $V_{v'}$  contient le plus de dominos sinon. Là encore, on choisit la plus petite valeur de  $v'$  s'il en existe plusieurs. Le domino sélectionné est alors retiré de  $V_v$  et  $V_{v'}$  (ou  $V_v$  seulement pour un domino  $\langle v : v \rangle$ ).

Les dominos suivants se trouvent en itérant le procédé : on prend comme domino suivant celui de  $V_{v'}$  qui est double ou qui porte la valeur  $v''$  pour laquelle  $V_{v''}$  contient le plus de domino. Et ainsi de suite.

Le train s'arrête quand on ne peut plus ajouter de dominos. Sauf cas exceptionnel, il reste des dominos inutilisés. Avec ces dominos inutilisés, on construit un deuxième train sur le même principe : on construit de nouveaux ensembles  $V_v$  avec les dominos inutilisés, on part d'une valeur  $v$  pour laquelle  $V_v$  est non vide, ... On répète le processus jusqu'à avoir utilisé tous les dominos pour construire des trains.

**Question 7** *Quelle est la longueur du plus long train construit de cette façon en prenant les dominos* **a)**  *dans  $D_{6,10}$  ?* **b)**  *dans  $D_{24,30}$  ?* **c)**  *dans  $D_{100,2000}$  ?*

**Question à développer pendant l'oral :** Quelle est la complexité en temps et en espace de votre algorithme ?





# Fiche réponse type: Dominos

$\widetilde{u}_0 : 1$

## Question 1

a)

b)

## Question 2

a)

b)

c)

## Question 3

a)

b)

## Question 4

a)

b)

c)

d)

## Question 5

a)

b)

c)

## Question 6

a)

b)

c)

## Question 7

a)

b)

c)







# Fiche réponse: Dominos

Nom, prénom, u<sub>0</sub>: .....

## Question 1

a)

b)

## Question 2

a)

b)

c)

## Question 3

a)

b)

## Question 4

a)

b)

c)

d)

## Question 5

a)

b)

c)

## Question 6

a)

b)

c)

## Question 7

a)

b)

c)



