



Concours National d'Informatique  
Sujet de demi-finale Paris 2

Samedi 10 mars 2007

# DES ÉCHELLES. DES SERPENTS.

## 1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre demi-finale. Sa durée est de 3 heures. Par la suite, vous passerez un entretien (20 minutes) et une épreuve de programmation sur machine (4 heures).

### Conseils

- Lisez bien tout le sujet avant de commencer.
- **Soignez la présentation** de votre copie.
- N'hésitez pas à poser des questions.
- Si vous avez fini en avance, relisez bien.
- N'oubliez pas de passer une bonne journée.

### Remarques

- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez la demi-finale et la date, en haut de votre copie.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le nous, nous pouvons ajouter des questions plus difficiles.
- Tous les langages sont autorisés. Néanmoins, veuillez préciser celui que vous utilisez.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe.

## 2 Sujet

### Introduction

Il fait noir, les murs étaient humides et Ulrich avançait péniblement dans le labyrinthe. Il tournait en rond depuis plusieurs heures et la seule source de lumière qu'il avait aperçue provenait d'en haut. « Il faut que je trouve un moyen de sortir de là » se répétait-il sans cesse. Il errait dans le dédale, sans savoir s'il allait du bon côté, mais il continuait d'avancer, évitant tant que possible les cadavres qui jonchaient le sol. Au bout d'un moment il entendit un bruit — des os, ou bien des pierres qui roulent ? — mais il choisit plutôt de l'ignorer. Il avança encore de quelques pas avant de sentir, de la main droite, du bois contre le mur. « Quelle chance, une échelle ! » Il y grimpa, mais il se retrouva dans un couloir identique. Il savait que la sortie était bien plus haut.

Après plusieurs minutes de marche, il perçut à nouveau le bruit. Cette fois il en était sûr, c'était le bruit de dés qu'on lance. D'où cela pouvait-il venir ? Ça ne pouvait pas être les rats... peut-être son imagination ? Non. Il avança encore, prudemment. Soudain, le sifflement d'un serpent se fit entendre. Il s'immobilisa de suite, s'attendant à voir quelque chose bouger. Il ne vit rien : pas de serpent, ni de rat ou quoi que soit de vivant. Il eut juste le temps de se rendre compte que le sol s'ouvrait sous pieds, avant de faire une chute d'au moins deux mètres. Tout le chemin serait à refaire...

### Le jeu

Ce jeu, absolument inintéressant et entièrement régi par le hasard, se joue sur un plateau où les cases sont numérotées, dans l'ordre. Il suffit de lancer à chaque tour un dé à six faces et d'avancer son personnage du même nombre de cases, dans le sens du labyrinthe. Lorsque l'on arrive sur une case en bas d'une échelle ou en haut d'un serpent, on doit déplacer son pion sur la case comme suit : Une échelle nous rapproche systématiquement de l'arrivée alors que le serpent l'en éloigne. En aucun cas, une échelle et un serpent ne pourront se trouver sur la même case (aussi bien pour le départ que pour l'arrivée).

Les pions commencent sur la première case. Le but du jeu est d'aller jusqu'à la sortie du labyrinthe qui est la dernière case du plateau. Lorsque le jet du dé est trop grand et demanderait à dépasser la sortie, le déplacement est annulé et le pion n'est pas déplacé.

Bref, c'est un jeu enfantin en somme, rien qui pourrait effrayer un brillant informaticien comme vous. Et pourtant...

### Question 1 (2 points)

Proposez une structure de données permettant de représenter un plateau de jeu. Votre représentation doit notamment décrire la position et l'orientation des échelles et des serpents. Expliquez et justifiez votre choix.

Attention, cette question est importante car vous utiliserez votre solution pour écrire les fonctions des questions suivantes. Relisez donc entièrement le sujet (et les questions qui suivent) avant de répondre.

### Question 2 (2 points)

À partir d'un plateau de jeu, de la position d'un pion et du résultat du dé, écrivez une fonction qui renvoie la position de la case d'arrivée. Votre fonction devra prendre en compte les différents éléments du jeu (échelles, serpents et fin du plateau).

### Question 3 (3 points)

Quelle est la case la plus proche de l'arrivée que l'on puisse atteindre en deux tours ? Écrivez une fonction qui calcule ce résultat à partir de la position actuelle d'un pion et d'un plateau.

## Question 4 (4 points)

On souhaite connaître le nombre de cases sur lesquelles on peut finir un tour. Ce n'est bien sûr pas possible de rester en bas d'une échelle ou en haut d'un serpent, mais il existe certainement des cases vides qui sont inaccessibles. Pensez notamment aux différentes configurations possibles et ce qu'il se passe quand il y a beaucoup d'échelles ou de serpents.

Écrivez une fonction qui renvoie cette valeur à partir d'un plateau donné en argument.

## Question 5

### Partie 1 (5 points)

Aidez Ulrich à sortir du labyrinthe le plus rapidement possible. Il a appris à éviter les pièges ; pour cette question, vous pouvez considérer que le plateau ne possède pas de serpents. Considérez aussi que les dés sont pipés (vous choisissez vous-même le résultat).

Écrivez une fonction qui renvoie le nombre de tours minimum pour permettre à Ulrich de sortir du labyrinthe (il est retourné sur la première case). Une part importante de la notation dépend de l'efficacité de votre algorithme.

### Partie 2 (2 points)

Par chance, Ulrich a retrouvé dans son sac le vieux boulier russe de son grand-père. Il souhaite exécuter votre algorithme sur le boulier. Il estime qu'il existe une centaine de cases dans le jeu, et environ 5 serpents et 5 échelles. Il espère de tout cœur que votre méthode est optimisée.

Calculez le temps que mettra Ulrich à exécuter votre algorithme. Il maîtrise l'art du boulier et peut effectuer jusqu'à dix opérations par seconde. De plus, il est très fort en mémorisation et peut retenir dix mille entiers, en utilisant des moyens mnémotechniques consistant à imaginer des recettes de cuisine. Par ailleurs, Ulrich a très faim.

### Partie 3 (0,12 point)

Ulrich survivra-t-il ? Vous ferez une analyse précise et détaillée du problème, en prenant en compte le temps de calcul nécessaire, le temps de faire le parcours, la faim d'Ulrich et le fait qu'utiliser un boulier dans le noir n'est pas trivial.

## Question 6 (bonus)

Reprenez votre solution de la question 5 et adaptez-la pour prendre en compte les serpents. Votre solution devra être optimisée. Cette question est plus dur qu'elle n'y paraît.

(+1 point présentation)