



Concours National d'Informatique

Sujet de demi-finale Paris I

23 janvier 2010

Maître de l'algorithmique

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).

Ceci est une épreuve d'algorithmique. Cela signifie que ce qui intéresse les correcteurs n'est pas la manière dont vous écrivez votre code, qui sera testée cet après-midi, mais votre manière de réfléchir et de résoudre des problèmes.

À ce titre, tous les langages sont autorisés, y compris le pseudo-code, pourvu que vous indiquiez lequel vous utilisez. Si vous éprouvez une quelconque difficulté avec votre langage, vous pouvez ainsi expliquer « en français » votre manière de résoudre la question, à condition que vous indiquiez un processus détaillé facilement transposable en un programme.

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.
- 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

Joseph Marchand est nul au Mastermind, ce jeu que vous connaissez très probablement où il s'agit de deviner un code de N pions ayant chacun une couleur parmi M couleurs possibles en faisant des propositions qui sont « corrigées » en indiquant le nombre de pions bien placés et le nombre de bonnes couleurs mal placées.

Il a donc décidé de tricher. Il va faire une application sur son iPhone qui calcule la meilleure stratégie pour jouer ! A chaque coup, elle lui indiquera le coup à jouer en fonction de la réponse qu'on lui a donnée. Ou plutôt, il vous demande de la faire pour lui, parce que même après toutes ces années à Prologin, il n'a pas appris grand chose...

Une *stratégie* est un arbre : un moment de la partie correspond à un nœud. Chaque nœud est étiqueté par la proposition que l'on va faire dans cette situation ; les branches qui partent de ce nœud sont étiquetées par les différentes réponses possibles à cette proposition.

Un exemple, pour $M = N = 2$, est donné en figure 1. La première ligne de chaque nœud donne les combinaisons encore possible, la deuxième celle qui est choisie.

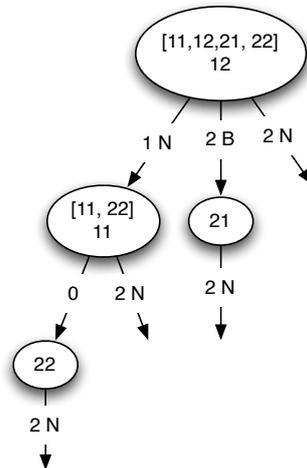


FIG. 1 : UNE STRATÉGIE POUR $N = M = 2$

Une stratégie sera optimale si, quand on l'applique à toutes les combinaisons en guise de code secret à la suite, elle génère le moins de coups au total. La stratégie donnée en exemple génère 3 coups si la solution est 22, 2 coups si la solution est 11 ou 21, un coup si c'est 12, soit huit coups au total.

Dans toute la suite du sujet, on aura $N \leq 4$ et $M \leq 8$.

Question 1 (1 point)

Proposez une structure de données pour stocker une proposition du joueur. Lisez la suite du sujet avant de répondre !

Question 2 (1 point)

Écrivez une fonction qui calcule le nombre de pions d'une couleur donnée dans une proposition.

Question 3 (2 points)

Le nombre de pions bien placés n'est pas ambigu ; il s'agit du nombre de pions de la même couleur que le pion à la même position dans la solution. Une fois ces pions retirés, on compte le nombre de pions mal placés. Si l'on trouve une paire de pions de la même couleur dans la proposition et la solution à des positions différentes, on compte une couleur mal placée et l'on répète le processus en ne considérant plus ces pions pour la suite des opérations.

Par exemple, si la proposition est 123 et la solution 112, il y a un pion bien placé (le 1) et un mal placé (le 2). Le 1 ne compte pas comme couleur mal placée. De même, si la proposition est 111 et que la solution est 123, la réponse est seulement « 1 bien placé », les autres 1 ne comptent pas. Enfin, si la proposition est 121 et la solution 212, on trouve 2 mal placés (l'un des 1 et le 2).

Écrivez une fonction qui donne le nombre de pions bien et mal placés, étant donné la proposition et la solution.

Question 4 (2 points)

Proposez une structure de données pour stocker une stratégie et un ensemble de solutions encore possibles. Lisez la suite du sujet avant de répondre...

Question 5 (3 points)

Écrivez une fonction qui calcule le nombre de coups générés au total par une stratégie que l'on vous passe en argument si on l'applique successivement à tous les codes possibles.

Question 6 (3 points)

Écrivez une fonction qui liste toutes les solutions possibles au début de la partie, étant donné N et M .

Question 7 (1 point)

Dans une stratégie optimale, est-il possible que l'on doive proposer une combinaison qui n'est pas possible étant donné les réponses que l'on nous a données précédemment ?

Question 8 (3 points)

Écrivez une fonction qui, étant donné un ensemble de solutions encore possibles et le code proposé par le joueur, crée les différents fils de ce nœud dans l'arbre de stratégie.

Question 9

Question 9a (3 points)

Écrivez une fonction qui renvoie l'arbre correspondant à une stratégie optimale.

Question 9b (1 point)

Quel est le temps d'exécution de votre algorithme pour $N = 4$ et $M = 6$ sur votre iPhone 3GS, équipé d'un processeur ARM à 600 MHz ?

Questions bonus

Question 10++

Proposez des pistes pour accélérer votre algorithme.

Question 11++

On vous donne une liste de coups et la réponse qui leur a été donnée. Écrivez un programme qui indique s'il existe ou non une solution (on sort ici du cadre du sujet ; il n'est plus question d'arbres de stratégie, juste d'une liste de coups et de réponses associées). Pensez-vous qu'il existe un algorithme polynomial pour résoudre ce problème ?

(+2 points offerts pour la présentation)