Jeu de MasterMind

Durée de l'épreuve : 4 heures — Coefficient : 4

Juillet 2003

Note. Lorsque la description d'un algorithme est demandée, celle-ci doit être courte et précise. Quand on demande la complexité d'un algorithme en fonction d'un paramètre n, on demande un ordre de grandeur du temps de calcul dans le cas le pire, par exemple $O(n^2)$ ou $O(n \log n)$.

Description du jeu de MasterMind. Étant donnés deux paramètres entiers $p \leqslant k$, on appelle MasterMind à p positions et k couleurs le jeu ci-dessous. Le premier joueur choisit en secret un p-uplet $s = (s[1], ..., s[p])$ d'éléments distincts dans l'ensemble des couleurs $\{1, ..., k\}$. L'autre joueur doit deviner s, et pour s'aider il pose des questions au premier joueur. Ces questions sont des p-uplets $q = (q[1], ..., q[p]) \in \{0, ..., k\}^p$ d'éléments non nécessairement distincts. La contrainte qui impose tous les $s[i]$ distincts sera appelée **propriété d'unicité dans le secret**. La possibilité de choisir des valeurs $q[i] = 0$, qui ne correspondent à aucune couleur, est l'autorisation d'utiliser des **questions partiellement vides**.
La réponse que fait le premier joueur est la paire d'entiers (N, B) définie par :
– N est le cardinal de $\{i | q[i] = s[i]\}$,
– $N + B$ est le cardinal de $\{i | \exists j : q[j] = s[i]\}$.
Le second joueur gagne s'il pose une question dont la réponse vérifie $N = p$. Le but de ce joueur est de minimiser le nombre de questions posées pour gagner.
NB : le véritable jeu de Master Mind™ a pour paramètres $p = 4$ et $k = 6$, avec des règles légèrement différentes de celles décrites ici. La propriété d'unicité dans le secret n'est pas vérifiée, et les questions partiellement vides ne sont pas autorisées (seules sont autorisées les **questions pleines**).

Les sections 2 et 3 peuvent être traitées indépendamment l'une de l'autre.

1 Préliminaires

On considèrera la suite d'entiers positifs (u_n) ci-dessous :
– u_0 est donné sur votre table. **(À reporter en haut de votre copie !)**
– $u_n = (16383 \times u_{n-1}) \bmod 59047$, pour $n > 0$.

Question 1 *Que valent u_{996} et u_{9996} ?*

Dans la suite de cette section, pour chaque question, on donnera la réponse pour chacun des paramètres suivants : $(p, k) = (4, 4)$, $(p, k) = (4, 5)$, $(p, k) = (4, 6)$, $(p, k) = (6, 8)$ et $(p, k) = (8, 10)$.

Pour $j > 0$, on définit la question $\kappa_j = (\kappa_j[1], ..., \kappa_j[p])$ par la formule $\kappa_j[i] = u_{i+j.p} \bmod (k + 1)$.

Question 2 *Quelles sont κ_1, κ_2 et κ_{10} ?*

On définit le secret $\sigma = (\sigma[1], ..., \sigma[p])$ par $\sigma[i] = 1 + v_{\lambda_i}$, où les λ_i sont définis par $\lambda_i = \min\left\{j \geqslant 0 | v_j \notin \{v_{\lambda_1}, ..., v_{\lambda_{i-1}}\}\right\}$ avec $v_j = u_j \bmod k$.

Question 3 *Que vaut σ ?*

La réponse à la question κ_j lorsque le secret est σ est la paire d'entiers notée (N_j^κ, B_j^κ).

Question 4 *Que valent (N_1^κ, B_1^κ), (N_2^κ, B_2^κ) et $(N_{10}^\kappa, B_{10}^\kappa)$?*

Question 5 *Parmi les 2000 premiers termes de la suite $(N_j^\kappa)_{j=1...2000}$, quelle est la valeur maximale ? Quel est le plus petit indice où cette valeur maximale est atteinte ?*

2 Recherche automatique du secret

On appliquera les méthodes de cette section aux paramètres $(p, k) = (4, 4)$, $(p, k) = (4, 5)$, $(p, k) = (4, 6)$ et $(p, k) = (6, 8)$.

Première méthode

Le second joueur essaie de deviner les couleurs $s[i]$ par ordre d'indice croissant, d'abord $s[1]$, puis $s[2]$, jusqu'à $s[p]$. Pour chaque indice, il essaie de deviner la couleur en testant toutes les valeurs par ordre croissant. Pour éviter un éventuel parasitage de la réponse, on utilisera des questions partiellement vides.

Question 6a *Décrivez un algorithme de recherche du secret qui met en pratique la méthode ci-dessus.*

Question 6b *Quel est le nombre de questions qu'il doit poser dans le cas le pire, en fonction de k et p.*

Question 6c *Pour σ, combien de questions pose-t-il pour gagner ?*

Deuxième méthode

Une amélioration est possible : au moment de deviner une valeur $s[i]$ avec $i > 1$, on ne pose pas certaines questions inutiles. Les questions inutiles sont celles qui testent une couleur, pour laquelle on peut déduire des réponses précédentes que $s[i]$ ne peut être cette couleur.

Question 7a *Décrivez ce qu'on peut déduire des réponses précédentes, et décrivez l'algorithme de recherche du secret.*

Question 7b *Quel est le nombre de questions qu'il doit poser dans le cas le pire, en fonction de k et p. On pourra se limiter au cas $k = p$.*

Question 7c *Pour σ, combien de questions pose-t-il pour gagner ?*

Troisième méthode

Cette méthode cherche à placer les couleurs dans leur position secrète, une par une, par numéro de couleur croissant. Pour chaque couleur on essaie toutes les positions non occupées jusqu'à trouver la bonne ; dès le premier essai, si la couleur n'est pas présente, elle est abandonnée.

Question 8a *Quel est le nombre de questions que doit poser cet algorithme dans le cas le pire, en fonction de k et p. On pourra se limiter au cas $k = p$.*

Question 8b *Pour σ, combien de questions pose-t-il pour gagner ?*

Question 9 *Comparez cet algorithme avec le précédent.*

Quatrième méthode

L'ensemble des questions possibles est naturellement muni de l'ordre lexicographique, défini par $q < q' \Leftrightarrow \sum_{i=1}^{p} q[i]k^{i-1} < \sum_{i=1}^{p} q'[i]k^{i-1}$.
On peut remarquer que les méthodes précédentes posent une suite de questions croissantes. On peut aussi remarquer que pour la plupart des questions posées la réponse $(p, 0)$ est impossible, c'est en particulier le cas pour les questions partiellement vides. Les méthodes précédentes peuvent donc vraisemblablement être améliorées...

Cette nouvelle méthode pose elle aussi des questions qui sont en ordre croissant, mais ne pose que des questions qui, compte-tenu des réponses précédentes, peuvent espérer recevoir la réponse $(N, B) = (p, 0)$.

Question 10 *Décrivez votre implantation de cette méthode. Pour σ, combien de questions pose-t-il pour gagner ? Quelles sont ces questions ?*

On mesure l'efficacité d'une méthode de résolution de MasterMind à l'aide de deux indicateurs :
- le nombre maximal de questions nécessaires pour gagner,
- le nombre moyen de questions (la moyenne étant calculée sur tous les secrets possibles, avec distribution uniforme).

Question 11 *Quelle est l'efficacité de cette méthode ?*

3 Choix de la question préférable

On appliquera les questions de cette section aux paramètres $(p, k) = (4, 4)$, $(p, k) = (4, 5)$, $(p, k) = (4, 6)$ et $(p, k) = (6, 8)$.

Question 12 *Quel est le nombre total de valeurs possibles pour le secret ?*

On appelle **historique** l'ensemble des connaissance du second joueur. C'est une séquence de questions $q_1 ... q_n$ avec les réponses reçues $(N_1, B_1) ... (N_n, B_n)$. L'historique est noté $H = \{(q_1, N_1, B_1), ..., (q_n, N_n, B_n)\}$.
Pour tout historique, le nombre de valeurs du secret qui sont compatibles avec les questions et réponses est noté $C[H]$.

Question 13a *Que vaut* $C[(\kappa_1, N_1^\kappa, B_1^\kappa)]$ *?*
Que vaut $C[(\kappa_1, N_1^\kappa, B_1^\kappa), ..., (\kappa_6, N_6^\kappa, B_6^\kappa)]$ *?*

Question 13b *Décrivez votre algorithme et évaluez sa complexité.*

On définit le **score maximal** de la question q compte tenu de l'historique H par la formule $S_M[q/H] = \max_{N,B} C[H, (q, N, B)]$.

Question 14 *Que vaut* $S_M[\kappa_1/\emptyset]$ *? Que vaut* $S_M[\kappa_2/(\kappa_1, N_1^\kappa, B_1^\kappa)]$ *?*

On définit le **score moyen** par la formule

$$S_\mu[q/H] = \frac{\sum_{N,B} C[H, (q, N, B)]^2}{\sum_{N,B} C[H, (q, N, B)]}.$$

Question 15 *Que vaut* $S_\mu[\kappa_1/\emptyset]$ *? Que vaut* $S_\mu[\kappa_2/(\kappa_1, N_1^\kappa, B_1^\kappa)]$ *?*

4 Algorithmes gloutons

On appliquera les méthodes de cette section aux paramètres $(p, k) = (2, 2)$, $(p, k) = (3, 3)$, $(p, k) = (3, 5)$, $(p, k) = (4, 4)$, $(p, k) = (4, 5)$.

Cinquième méthode

On considère ici un algorithme glouton, qui choisit à chaque étape une question qui minimise le nombre de possibilités restantes, dans le cas le pire.
Il s'agit donc de choisir parmi toutes les questions possibles (y compris celles qui sont partiellement vides), l'une de celles dont le **score maximal** est minimal. On choisira par exemple la première de ces questions selon l'ordre lexicographique.
Remarquons qu'il est possible qu'après avoir posé un certain nombre de questions, il soit possible de connaître le secret (seul compatible) sans avoir gagné (si le secret n'a pas été posé comme question).

Question 16 *Pour* σ*, combien de questions suffit-il de poser pour connaître le secret ? Combien de questions doit-on poser pour gagner avec cette méthode ? Quelles sont ces questions ?*

Question 17 *Quelle est l'efficacité de cette méthode ? Comparez aux méthodes précédentes.*

Sixième méthode

C'est une variante de la méthode précédente. On choisit une question de **score maximal** minimal, parmi les questions dont tous les $q[i]$ sont non nuls et distincts.

Question 18 *Pour σ, combien de questions suffit-il de poser pour connaître le secret ? Combien de questions doit-on poser pour gagner avec cette méthode ? Quelles sont ces questions ?*

Question 19 *Quelle est l'efficacité de cette méthode ? Comparez aux méthodes précédentes.*

Septième et huitième méthodes

Ce sont des variantes des deux méthodes précédentes, où la meilleure question est choisie en minimisant le **score moyen** au lieu du **score maximal**.

Question 20 *Pour σ, combien de questions suffit-il de poser pour connaître le secret ? Combien de questions doit-on poser pour gagner avec chacune de ces méthodes ? Quelles sont ces questions ?*

Question 21 *Quelle est l'efficacité de chacune de ces méthodes ? Comparez aux méthodes précédentes.*