

TP ENS : Awale

Pour ce TP type ENS, on traitera les questions de programmation 1, 2, 3 et 4 en langage OCaml, les questions 5, 6 et 7 en langage C, et les questions 8, 9, 10, 11, 12 dans le langage de son choix.

Pour les questions 5, 6 et 7, on téléchargera l'archive `TPENS_awale.zip` sur l'ENT. Cette archive contient trois fichiers :

- deux fichiers textes contenant les configurations de jeu pour la valeur $\tilde{u}_0 = 42$ de référence ;
- un fichier `awale.c` permettant de lire ces fichiers texte pour interpréter les valeurs de configuration. Ainsi, après la lecture des fichiers, si `n` est une valeur entière, alors `Dw7[n]` renvoie un tableau d'entiers correspondant à $\mathcal{D}(w_{7,n})$ et `Dw24[n]` renvoie un tableau d'entiers correspondant à $\mathcal{D}(w_{24,n})$.

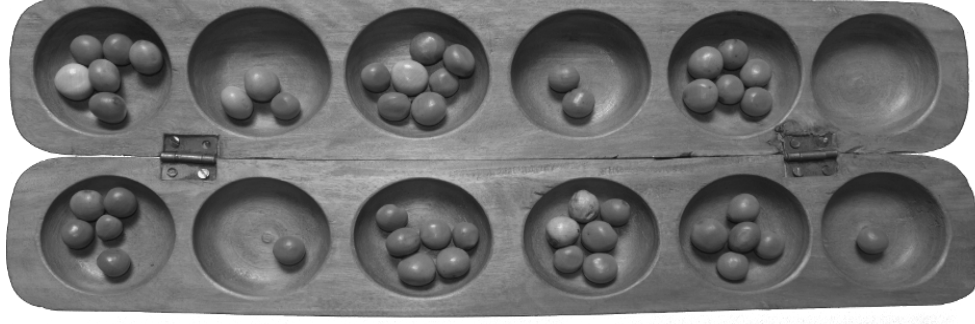
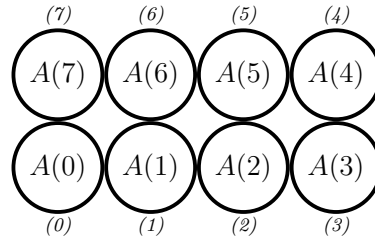


FIGURE 1 – 6-Awalé

1 Introduction

L'Awalé est un jeu combinatoire à information parfaite dont l'origine est généralement associée à la fédération Ashanti, en Afrique occidentale. Dans ce sujet, nous suivrons les règles pratiquées dans la ville de Tamale (Ghana) par les joueurs se réunissant à l'angle des rues Salaga et Nyohini.

L'Awalé se joue traditionnellement sur un plateau composé de 6 puits pour le joueur et 6 puits pour son adversaire (figure 1). Cependant, on se limitera ici pour simplifier à l'étude du 4-Awalé, c'est-à-dire les Awalés de 4 puits par côté. Le jeu comporte initialement 3 graines par puits, soit 24 graines au total, et évolue ensuite par déplacement et capture des graines. Symboliquement, on représentera un état du jeu de la façon suivante



Par convention, on placera le côté joueur (dont c'est le tour de jouer) en bas et le côté adverse (qui vient de jouer) en haut. Le jeu étant symétrique, cette représentation est canonique et est appelée une configuration. Une configuration contenant exactement i graines est appelée une i -configuration. On utilise la notation $A(i)$ pour indiquer le nombre de graines dans le puits $i \in [0, 7]$.

Dans la suite, on note $a \bmod b$ le reste de la division euclidienne de a par b . On note $\lfloor x \rfloor$ la partie entière par défaut. Finalement, on note $\binom{n}{k}$ la combinaison de k parmi n .

Génération pseudo-aléatoire. À partir de la valeur initiale u_0 qui vous a été donnée, on définit cinq familles d'entiers $(u_n)_{n \geq 0}$, $(v_{7,n})_{n \geq 0}$, $(v_{24,n})_{n \geq 0}$, $(w_{7,n})_{n \geq 0}$ et $(w_{24,n})_{n \geq 0}$ par :

$$\begin{array}{lcl}
 u_{n+1} = (au_n + c) \bmod m & & \\
 v_{7,n} = \lfloor 3432 \times \frac{u_n}{m} \rfloor & w_{7,n} = 7 + 25 \times v_{7,n} & \text{avec} \\
 v_{24,n} = \lfloor 2629575 \times \frac{u_n}{m} \rfloor & w_{24,n} = 24 + 25 \times v_{24,n} & \left\{ \begin{array}{l} a = 1103515245 \\ c = 12345 \\ m = 2^{31} \end{array} \right.
 \end{array}$$

Question 1 Calculer :

- a) $(u_1 \bmod 37, v_{7,1} \bmod 37, w_{7,1} \bmod 37)$,
- b) $(u_{1000} \bmod 37, v_{24,1000} \bmod 37, w_{24,1000} \bmod 37)$,
- c) $(u_{1000000} \bmod 37, v_{24,1000000} \bmod 37, w_{24,1000000} \bmod 37)$

Indication. On veillera à éviter les dépassements d'entiers lors du calcul de u_n .

Coefficients binomiaux. Dans la suite du sujet, il est fréquemment fait usage des coefficients binomiaux $\binom{j}{i}$ pour $1 \leq i \leq 7$ et $i - 1 \leq j \leq 24 + i$.

Question 2 Calculer $\binom{j}{i}$ pour :

- a) $i = 4 + (u_{10} \bmod 4)$, $j = i + 5 + (u_{11} \bmod 20)$
- b) $i = 4 + (u_{12} \bmod 4)$, $j = i + 5 + (u_{13} \bmod 20)$
- c) $i = 4 + (u_{14} \bmod 4)$, $j = i + 5 + (u_{15} \bmod 20)$

Indication. Afin d'éviter les dépassements d'entiers lors du calcul des coefficients, on pourra tirer parti de la formule de Pascal

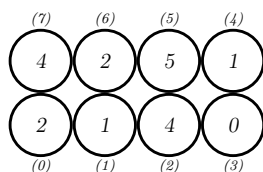
$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

2 Codage de Gödel

On établit maintenant un codage des configurations par des entiers naturels. Cette représentation compacte sera utile dans la formulation des questions posées dans la suite du sujet, mais ce n'est généralement pas une bonne structure de données pour les calculs. Considérons tout d'abord l'encodage et le décodage des n -configurations (comptant donc n graines), pour un n fixé. On remarque qu'une n -configuration A est caractérisée par un vecteur $\vec{c} = (c_0, \dots, c_6)$ avec $0 \leq c_0 < c_1 < \dots < c_6 < n + 7$, donné par

$$c_0 = A(0), \quad c_{k+1} = A(k+1) + 1 + c_k.$$

Exemple 1 À l'Awalé



correspond le vecteur $(2, 4, 9, 10, 12, 18, 21)$.

Question à développer pendant l'oral 1 Exprimer, en fonction de $n \in [0, 24]$, le nombre de n -configurations possibles.

Étant donné un tel vecteur \vec{c} , on pose

$$enc(\vec{c}) = \sum_{i=1}^7 \binom{c_{i-1}}{i}.$$

Inversement, à un entier g , on associe le vecteur $dec(g) = (c_0, \dots, c_6)$ où :

$$\begin{aligned} c_6 &\text{ est le plus grand entier vérifiant } \binom{c_6}{7} \leq g_7 \text{ avec } g_7 = g \\ c_5 &\text{ est le plus grand entier vérifiant } \binom{c_5}{6} \leq g_6 \text{ avec } g_6 = g_7 - \binom{c_6}{7} \\ &\vdots \\ c_0 &\text{ est le plus grand entier vérifiant } \binom{c_0}{1} \leq g_1 \text{ avec } g_1 = g_2 - \binom{c_1}{2} \end{aligned}$$

Question 3 Calculer :

a) $dec(v_{7,1})$, **b)** $dec(v_{7,2})$, **c)** $dec(v_{7,3})$, **d)** $dec(v_{24,4})$, **e)** $dec(v_{24,5})$, **f)** $dec(v_{24,6})$

On pourra vérifier numériquement sur quelques exemples que les fonctions enc et dec sont inverses l'une de l'autre. (Voir question à développer pendant l'oral 4 ci-dessous.)

Considérons maintenant une configuration A contenant un nombre n quelconque de graines. On appelle nombre de Gödel de A la quantité

$$\mathcal{E}(A) = n + 25 \times enc(\vec{c})$$

où \vec{c} est le vecteur de coefficients associé à A .

Question à développer pendant l'oral 2 Quel est le plus grand nombre de Gödel g_{\max} représentant une configuration valide comportant au plus 24 graines ? Est-ce qu'un tel entier est représentable sur un mot machine de 64 bits ?

Étant donné un nombre de Gödel g , on retrouve d'abord le nombre de graines correspondant en calculant $n = g \bmod 25$. À partir de ce nombre et du vecteur de coefficients $dec(\lfloor g/25 \rfloor)$, il est aisé de reconstruire la configuration de nombre de Gödel g . On note $\mathcal{D}(g)$ cette configuration.

Question 4 Calculer :

a) $\mathcal{D}(w_{7,1})$, **b)** $\mathcal{D}(w_{7,2})$, **c)** $\mathcal{D}(w_{7,3})$, **d)** $\mathcal{D}(w_{24,4})$, **e)** $\mathcal{D}(w_{24,5})$, **f)** $\mathcal{D}(w_{24,6})$

Injectivité du codage de Gödel. Dans le reste de cette section, on montre que enc forme une bijection. Ce résultat n'est pas nécessaire pour traiter le reste du sujet.

Soient t et u des entiers. On note $V_{t,u}$ l'ensemble des vecteurs (c_0, \dots, c_{t-1}) avec $0 \leq c_0 < c_1 < \dots < c_{t-1} < u$. On ordonne les éléments de $V_{t,u}$ par ordre lexicographique où les coefficients sont lus à partir de la fin, c'est-à-dire que l'on convient que $\vec{c} < \vec{d}$ s'il existe un indice k tel que

$$\begin{cases} \forall i \in [k+1, t-1], c_i = d_i \\ c_k < d_k \end{cases}$$

Exemple 2 Les éléments de $V_{3,5}$ sont, dans l'ordre :

$$\begin{aligned} (0, 1, 2) &< (0, 1, 3) < (0, 2, 3) < (1, 2, 3) < (0, 1, 4) \\ &< (0, 2, 4) < (1, 2, 4) < (0, 3, 4) < (1, 3, 4) < (2, 3, 4). \end{aligned}$$

Question à développer pendant l'oral 3 Lorsque l'on énumère les éléments de $V_{t,u}$ dans l'ordre défini ci-dessus, montrer que le vecteur (c_0, \dots, c_{t-1}) est visité après avoir énuméré exactement

$$\binom{c_0}{1} + \binom{c_1}{2} + \dots + \binom{c_{t-1}}{t}$$

vecteurs.

Question à développer pendant l'oral 4 Montrer que

$$\binom{c_{t-1}}{t} \leq \sum_{i=1}^t \binom{c_{i-1}}{i} < \binom{c_{t-1} + 1}{t}.$$

En conclure que pour n fixé, les fonctions d'encodage enc et de décodage dec sont inverses l'une de l'autre.

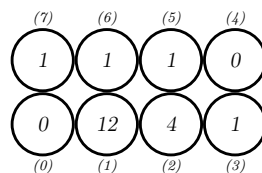
3 Semaines et récoltes

Les joueurs jouent chacun leur tour. Le tour d'un joueur comporte deux temps :

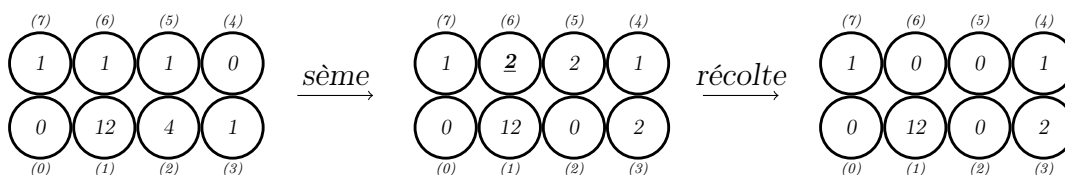
1. semaillages : le joueur courant choisit un puits non vide $\langle p \rangle$ de son camp, se saisit des graines contenues dans ce puits, et les dépose (les "sème") une par une dans les *autres* puits, dans le sens anti-horaire, en commençant par le puits suivant $\langle (p+1) \bmod 8 \rangle$. On ne sème jamais dans le puits d'origine : si l'on revient au puits $\langle p \rangle$, on le passe sans y déposer de graine et l'on continue à semer les graines restantes à partir du puits suivant ;
2. récolte : si la dernière graine semée l'a été dans un puits adverse $\langle q \rangle$ et si le nombre de graines dans ce puits satisfait la condition de capture $2 \leq A(q) \leq 3$, le joueur gagne (les "récolte") les graines du puits $\langle q \rangle$ et les met de côté. Il récolte de même les graines des puits précédents $\langle q' \rangle = \langle q-1 \rangle, \langle q-2 \rangle, \dots$ jusqu'à arriver à un puits $\langle q_f \rangle$ appartenant à son propre camp ou ne satisfaisant pas la condition de capture.

On note $coup(m, A)$ la configuration obtenue après avoir semé les graines du puits m de la configuration A et effectué la récolte. On note $gain(m, A)$ le nombre de graines récoltées par l'opération $coup(m, A)$.

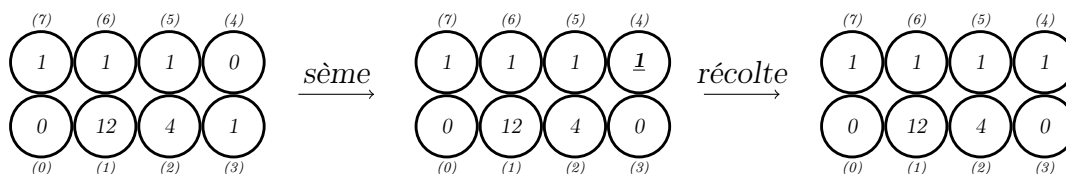
Exemple 3 Considérons l'Awalé



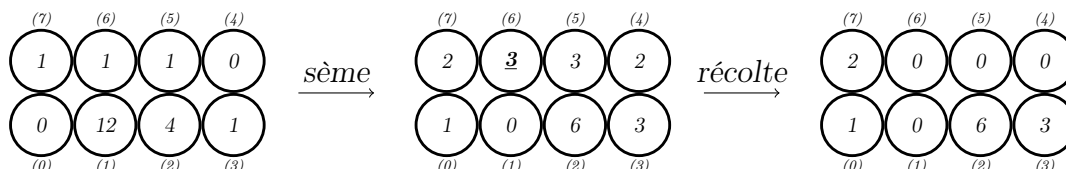
Si l'on sème les graines du puits $\langle 2 \rangle$, on récolte 4 graines par l'opération suivante :



Si, par contre, on sème les graines du puits $\langle 3 \rangle$, on ne récolte aucune graine :

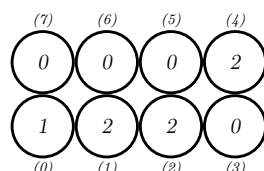


Si, enfin, on sème les graines du puits $\langle 1 \rangle$, on récolte 8 graines :



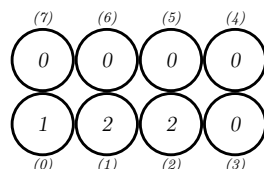
De plus, le joueur ne peut jouer que les coups qui laissent la possibilité à son adversaire de poursuivre la partie. On dira d'un joueur ne disposant d'aucune graine qu'il est en situation de famine. Ainsi, un puits est jouable s'il contient au moins une graine et que l'adversaire n'est pas en situation de famine suite à ce coup. On note $\text{jouable}(A)$ l'ensemble des puits jouables par le joueur de la configuration A . Une configuration est dite terminale (du point de vue du joueur) si le joueur ne dispose d'aucun puits jouable. Dans ce cas, l'adversaire remporte toutes les graines présentes sur le plateau.

Exemple 4 Considérons l'Awalé



Les puits $\langle 2 \rangle$ et $\langle 3 \rangle$ ne sont pas jouables : le second ne contient aucune graine tandis que le premier entraîne la famine de l'adversaire. Les autres puits sont jouables.

Considérons l'Awalé



où l'adversaire s'est mis en situation de famine en jouant $\langle 7 \rangle$ au coup précédent. Le puits $\langle 2 \rangle$ est le seul puits jouable : les autres puits sont soit vides, soit ils maintiennent l'adversaire dans son état de famine.

Question 5 Calculer les trois premières valeurs $1 \leq n_0 < n_1 < n_2$ pour lesquelles le joueur des configurations $\mathcal{D}(w_{7,n_0})$, $\mathcal{D}(w_{7,n_1})$ et $\mathcal{D}(w_{7,n_2})$ ne peut jouer aucun puits bien qu'il dispose de graines : **a)** $n_0 = ?$, **b)** $n_1 = ?$, **c)** $n_2 = ?$

Question 6 Pour $0 \leq i \leq 4$, calculer le plus petit indice $n_i \geq 1$ pour lequel la configuration $\mathcal{D}(w_{24,n_i})$ dispose d'exactly i puits jouables. Pour chacun des $i > 0$, calculer le nombre maximal de graines r_i pouvant être récoltées en jouant l'un de ces puits :

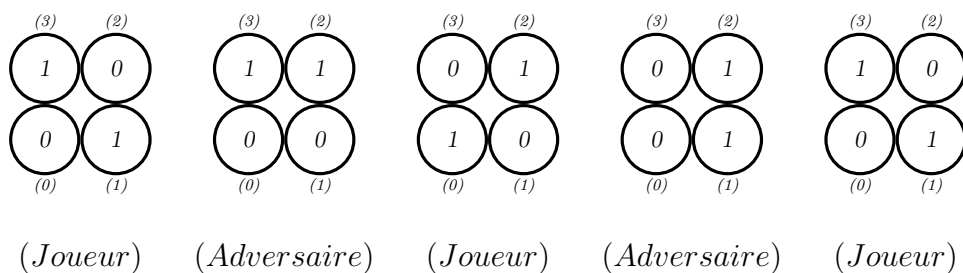
a) $n_0 = ?$, **b)** $(n_1, r_1) = ?$, **c)** $(n_2, r_2) = ?$, **d)** $(n_3, r_3) = ?$, **e)** $(n_4, r_4) = ?$

Question 7 Calculer les trois premières valeurs de $1 \leq n_0 < n_1 < n_2$ pour lesquelles les configurations $\mathcal{D}(w_{24,n_0})$, $\mathcal{D}(w_{24,n_1})$ et $\mathcal{D}(w_{24,n_2})$ sont terminales :

a) $n_0 = ?$, **b)** $n_1 = ?$, **c)** $n_2 = ?$

Il est possible pour les joueurs de retourner dans une configuration déjà jouée par le passé. Si, durant le jeu, une configuration est répétée, c'est un mat : chaque joueur s'empare de la moitié (entière) des graines restantes.

Exemple 5 Sur un 2-Awalé, la séquence de coup $(\langle 1 \rangle, \langle 3 \rangle, \langle 0 \rangle, \langle 2 \rangle)$ donne lieu à un mat :



Score. Jusqu'à présent, nos définitions se sont concentrées sur les déplacements du joueur, au détriment de ceux de l'adversaire. Le jeu étant symétrique, on obtient les opérations adaptées à l'adversaire en effectuant une rotation de la configuration : le joueur passe en position d'adversaire et inversement. On notera $rotation(A)$ la configuration obtenue par rotation de A .

Question 8 Calculer le nombre de Gödel de la rotation de chacune des configurations suivantes : **a)** $\mathcal{D}(w_{7,30})$, **b)** $\mathcal{D}(w_{24,31})$, **c)** $\mathcal{D}(w_{24,32})$

On définit le score d'une configuration comme la différence entre le nombre de graines que le joueur peut espérer gagner et le nombre de graines que son adversaire peut espérer gagner, chacun jouant de façon optimale (et en supposant que cette configuration n'est pas apparue auparavant dans la partie).

Ainsi, le score d'une configuration terminale est l'opposé du nombre de graines gagnées par l'adversaire, et le score d'une configuration non terminale vérifie

$$score(A) = \max_{m \in jouable(A)} (gain(m, A) - score(rotation(coup(m, A)))).$$

Question à développer pendant l'oral 5 Si le score de la configuration initiale du 4-Awalé est strictement positif, que peut-on en conclure ? Et si celui-ci est strictement négatif ? Est-il possible que le score de la configuration initiale soit nul ?

Question à développer pendant l'oral 6 Dans le pire cas, en supposant que tous les puits sont jouables et donnent des configurations distinctes, déterminer le nombre de configurations accessibles après l coups alternants entre joueur et adversaire.

Pourrait-on déterminer le score de la configuration initiale du 4-Awalé par une simple exploration en profondeur ?

4 Analyse rétrograde

L'analyse rétrograde, inventée par R. Bellman, utilise la technique de programmation dynamique pour établir le score de l'ensemble des $(n + 1)$ -configurations à partir du score des k -configurations pour $k \leq n$. Pour ce faire, l'algorithme est initialisé avec l'ensemble des $(n + 1)$ -configurations terminales puis procède par anti-coups : à partir d'une configuration donnée, on énumère toutes les configurations ayant pu mener à celle-ci sans récolte.

Anti-coups. Étant donné une n -configuration A , on se propose de calculer l'ensemble E des n -configurations tel que, pour tout $A' \in E$, il existe un coup jouable dans A' menant à la configuration A . On s'intéresse ainsi aux semailles ne débouchant sur aucune récolte et on ne considère pas les n' -configurations (avec $n' > n$) qui mènent à A suite à une récolte fructueuse. On dira que A' est obtenu par anti-coup depuis la configuration A .

Question 9 Pour $0 \leq k \leq 6$, calculer le plus petit indice $n_k \geq 1$ pour lequel la configuration $\mathcal{D}(w_{24,n_k})$ dispose de k configurations accessibles par anti-coups. On notera s_k la somme modulo 37 des encodages de Gödel de ces configurations :

a) $(n_0, s_0) = ?$, **b)** $(n_1, s_1) = ?$, **c)** $(n_2, s_2) = ?$, **d)** $(n_3, s_3) = ?$,
e) $(n_4, s_4) = ?$, **f)** $(n_5, s_5) = ?$, **g)** $(n_6, s_6) = ?$

Analyse rétrograde. L'analyse rétrograde procède par itération sur le nombre de graines $n \in [0, 24]$. Chaque itération calcule une base de données associant un score à chacune des n -configurations. Une itération se compose de 3 phases :

1. Initialisation, où la base de données des n -configurations est initialisée avec les configurations terminales et les configurations avec récolte.
2. Convergence, où l'on propage les scores aux configurations parentes par anti-coups.
3. Stabilisation, où l'on identifie et attribue un score aux situations de mat.

Durant la phase d'initialisation et de propagation, une configuration peut être dans l'un des deux états suivants :

Instable(t, k) : une borne inférieure t du score final est connue mais k configurations filles n'ont pas encore informé le nœud de leurs scores respectifs ;

Stable(s) : le score final s est connu et a été propagé.

Phase 1 : initialisation. L'initialisation consiste à itérer sur l'ensemble des n -configurations. Pour les configurations non terminales, les puits jouables se scindent en deux parties :

- ceux qui donnent lieu à une récolte de k graines, dont le score s de la configuration résultante a été pré-calculé dans une base de données indexant les configurations de $n - k$ graines. Le score du coup est alors de $k - s$;
- ceux qui ne font aucune récolte, dont la configuration résultante appartient à la base de données en cours de construction et dont le score est encore inconnu.

Les configurations non terminales sont donc instables. Les coups récoltants permettent d'estimer la borne inférieure du score de la configuration tandis que le nombre de coups non récoltants indique le nombre de configurations filles restant à déterminer.

Les configurations terminales sont un cas particulier : elles sont instables de score $-n$ mais en attente de 0 configuration fille. La phase suivante aura pour effet de propager leur score.

Question 10 *Considérons un jeu de bases de données tests pour les configurations de taille $n \in [0, 6]$ où chaque n -configuration A est marquée comme stable de score $(u(\mathcal{E}(A)) \bmod (2n + 1)) - n$.*

À partir de ces bases de données, effectuer l'initialisation de la base de données des 7-configurations puis, pour $0 \leq k \leq 4$, calculer le plus petit indice $n_k \geq 1$ pour lequel la configuration $\mathcal{D}(w_{7,n_k})$ a été initialisée dans un état instable avec exactement k configurations filles manquantes. Soit s_k l'estimation du score de $\mathcal{D}(w_{7,n_k})$ à son initialisation.

a) $(n_0, s_0) = ?$, **b)** $(n_1, s_1) = ?$, **c)** $(n_2, s_2) = ?$, **d)** $(n_3, s_3) = ?$, **e)** $(n_4, s_4) = ?$

Phase 2 : convergence. La convergence effectue $n + 1$ itérations sur l'ensemble des n -configurations. Lors de la $l^{\text{ème}}$ itération (pour $l \in [0, n]$), on dira d'une configuration dans un état instable qu'elle est l -saturée si toutes ses configurations filles lui ont propagé leurs scores ou que son score est égal à $n - l$. Durant cette $l^{\text{ème}}$ itération, chaque configuration l -saturée est marquée comme stable et son score est propagé récursivement à ses configurations parentes par anti-coups.

La propagation se déroule en deux phases. Tout d'abord, on calcule les parents (sans récolte) de la configuration. On notifie ensuite chacun des parents du score de la configuration fille. Si cette information permet de saturer la configuration parent alors celle-ci est marquée comme stable et son score est récursivement propagé. Le cas échéant, on met à jour le score et le nombre de configurations filles restant à déterminer.

Question 11 *À partir de la base de données initialisée en question 10, calculer la paire du nombre d'itérations et du score auxquelles les configurations suivantes passent d'un état instable à un état stable durant la phase de convergence. On notera "mat" les configurations qui n'auraient pas atteint un état stable à la fin cette phase.*

a) $\mathcal{D}(w_{7,80})$ **b)** $\mathcal{D}(w_{7,81})$ **c)** $\mathcal{D}(w_{7,82})$ **d)** $\mathcal{D}(w_{7,83})$ **e)** $\mathcal{D}(w_{7,84})$ **f)** $\mathcal{D}(w_{7,85})$

Indication. On pourra vérifier la cohérence des scores obtenus en comparant avec le score calculé à partir des configurations filles (selon la formule introduite en partie 3). Attention, ce test n'apporte pas de garantie quant à la correction des bases de données, il ne fait que tester l'absence d'erreurs grossières.

Question à développer pendant l'oral 7 *Établir une borne supérieure sur la valeur absolue du score des positions marquées comme stables durant la $l^{\text{ème}}$ itération.*

En déduire que le critère de saturation est justifié en montrant qu'une configuration l -saturée lors de la $l^{\text{ème}}$ itération a effectivement atteint son score maximal.

Que dire des configurations qui restent dans un état instable à l'issue de la $n + 1^{\text{ème}}$ itération ?

Phase 3 : stabilisation. La phase de stabilisation itère donc une dernière fois sur les entrées de la base de données : les configurations encore instables sont marquées stables, de score nul.

Question 12 Effectuer l'analyse rétrograde pour le 4-Awalé. À l'aide des bases de données obtenues, calculer le score des configurations suivantes :

a) $\mathcal{D}(w_{24,90})$, b) $\mathcal{D}(w_{24,91})$, c) $\mathcal{D}(w_{24,92})$

Question à développer pendant l'oral 8 Est-ce que le joueur de 4-Awalé ou son adversaire dispose d'une stratégie gagnante ?

Quelle est la complexité temporelle de votre analyse rétrograde, toutes phases comprises, pour le calcul de la base de données des n -configurations ? Quel sera l'espace mémoire occupé par l'ensemble des bases de données ?

Question à développer pendant l'oral 9 Concernant la complexité spatiale, pensez-vous que l'on puisse efficacement allouer la mémoire nécessaire à l'exécution de l'algorithme pour un 6-Awalé de 48 graines sur

1. la mémoire vive ?
2. un disque dur ?

En extrapolant à partir de votre traitement du 4-Awalé, donnez une estimation du temps d'exécution de l'analyse rétrograde pour le 6-Awalé.



Fiche réponse type: Le jeu d'Awalé

\widetilde{u}_0 : 42

Question 1

- a)
- b)
- c)

Question 2

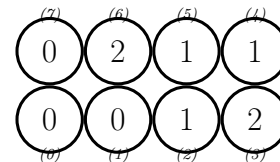
- a)
- b)
- c)

Question 3

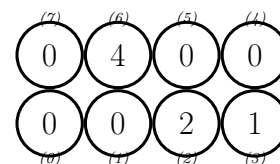
- a)
- b)
- c)
- d)
- e)
- f)

Question 4

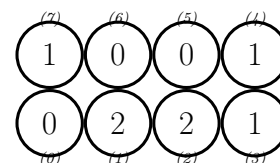
a)



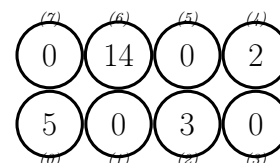
b)



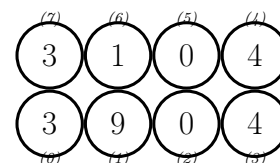
c)

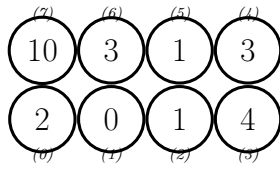


d)



e)





f)

Question 5

a)

235

b)

290

c)

309

Question 6

a)

1816

b)

(26, 0)

c)

(4, 0)

d)

(1, 0)

e)

(7, 3)

Question 7

a)

1816

b)

1888

c)

3663

Question 8

a)

69232

b)

46896199

c)

46310774

Question 9

a)

(7, 0)

b)

(1, 36)

c)

(2, 3)

d)

(33, 30)

e)

(8, 31)

f)

(6, 0)

g)

(16, 11)

Question 10

a)

(16, 6)

b)

(1, 4)

c)

(2, -7)

d)

(6, -7)

e)

(77, -7)

Question 11

a)

(6, -1)

- b)
- c)
- d)
- e)
- f)

Question 12

- a)
- b)
- c)

