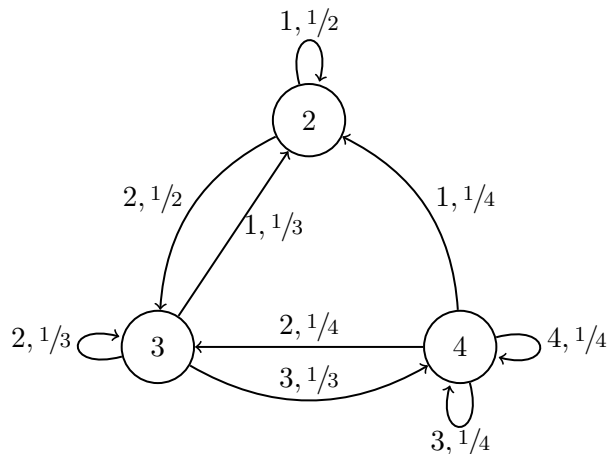


Solutionnaire du travail pratique #1

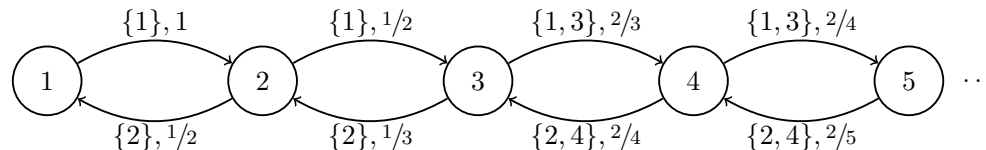
Réponses

Note: les réponses aux différentes questions ne sont pas toujours uniques.

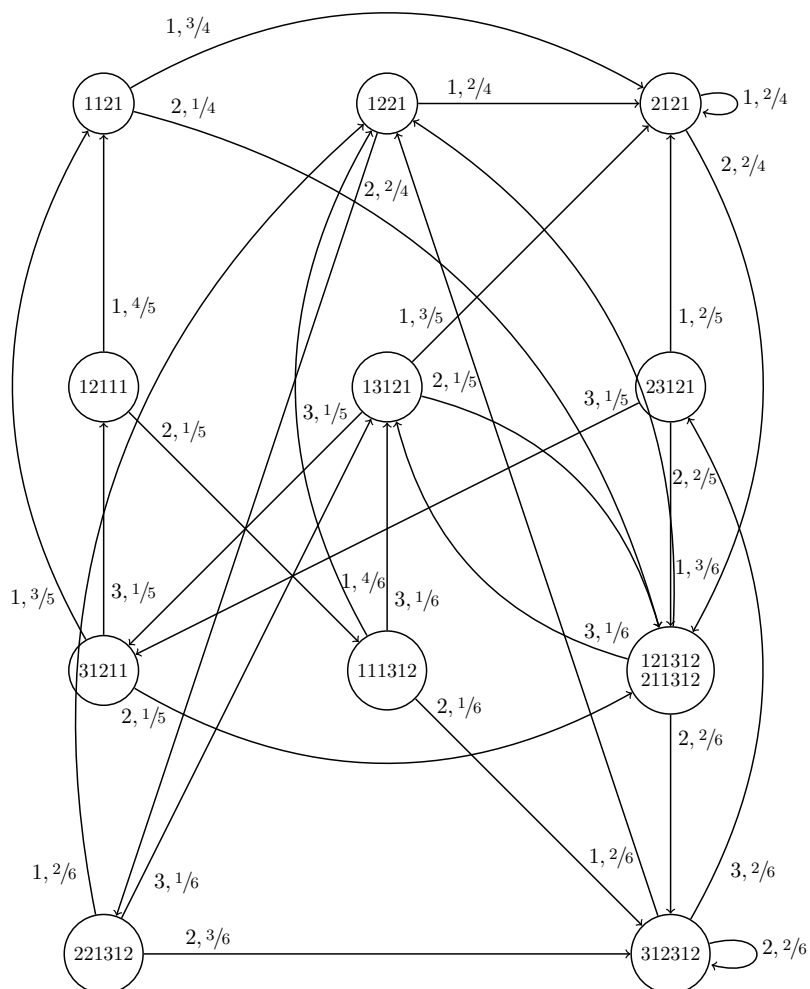
- (a) Chaque état de l'automate indique combien de faces le dé courant possède. Donc, on se trouve à l'état 2 si on utilise le dé Δ_1 , à l'état 3 si on utilise le dé Δ_2 et à l'état 4 si utilise le dé Δ_3 ou le dé Δ_4 . J'ai étiqueté les transitions avec les faces qu'on peut obtenir avec le dé lancé ainsi qu'avec les probabilités correspondantes. Le modèle est d'ordre 1 car le tout dernier tirage d'un dé détermine complètement l'état courant.



- (b) Le numéro k de chaque état de l'automate est l'une des valeurs possibles de la variable N_i , pour $i \geq k$. Les événements possibles lors du tirage fait dans l'état k sont $1, 2, \dots, k$. J'ai étiqueté chaque transition avec un ensemble de faces qu'on peut obtenir avec le dé lancé ainsi qu'avec la probabilité de tomber sur un numéro faisant partie de cet ensemble. Le modèle est d'ordre infini car il faut se souvenir de tous les tirages depuis le début pour connaître avec certitude l'état courant.



- (c) Chaque état de l'automate correspond à une clé spécifique qu'il est possible d'obtenir. Il y a un état qui correspond à deux clés à la fois: ceci vient du fait le tirage et les circonstances futures sont les mêmes, que ce soit avec une clé ou avec l'autre. J'ai étiqueté les transitions avec les chiffres et les probabilités correspondantes. Le modèle est d'ordre 4 car le fait de connaître les quatre derniers chiffres suffit à déterminer sans aucun doute l'état courant; voir le tableau qui se trouve sous l'automate. Le modèle n'est pas d'ordre 3 car, après les chiffres 332, on n'est sûr de l'état atteint.



Récents: état	Récents: état	Récents: état
11 : 2121	312 : 121312/211312	13 :
21 : 1221	22 : 312312	123 : 13121
131 :	132 :	223 : 23121
231 : 2121	232 : 121312/211312	323 : 13121
331 : 1121	1332 :	133 :
112 : 121312/211312	2332 : 121312/211312	233 : 31211
212 : 221312	3332 : 111312	333 : 12111

2. (a) Un tirage du dé Δ_n peut être simulé à l'aide de variables aléatoires plus simples. Premièrement, une pièce de monnaie fautive sert à déterminer si on opte pour la face préférée de Δ_n ou pas. Deuxièmement, si ce n'est pas la face préférée qui est choisie, un dé équilibré à $n - 1$ faces sert à déterminer quelle face ordinaire de Δ_n est choisie. La pièce fautive donne "préférée" avec probabilité p_n et "ordinaire" avec probabilité $1 - p_n$. Donc, l'entropie¹ de Δ_n est:

$$H(\Delta_n) = p_n \log \frac{1}{p_n} + (1 - p_n) \left(\log \frac{1}{1 - p_n} + \log(n - 1) \right) .$$

De plus, selon l'hypothèse donnée dans l'énoncé du TP, $H(\Delta_n) = 1$. Or, à cause de la forme complexe de la formule pour $H(\Delta_n)$, il n'est pas possible d'isoler la variable p_n , en général. Le cas de Δ_2 est spécial et il peut être résolu exactement mais les autres cas nécessitent une résolution numérique. Dans le cas de Δ_2 , on a:

$$\begin{aligned} 1 &= H(\Delta_2) \\ &= p_2 \log \frac{1}{p_2} + (1 - p_2) \left(\log \frac{1}{1 - p_2} + \log(2 - 1) \right) . \\ &= p_2 \log \frac{1}{p_2} + (1 - p_2) \log \frac{1}{1 - p_2} . \end{aligned}$$

L'imprévisibilité du dé Δ_2 se résume à l'imprévisibilité de la pièce de monnaie hypothétique utilisée dans notre raisonnement ci-haut. Or, le seul moyen d'avoir une entropie de 1 pour une pièce de monnaie est d'avoir deux faces équiprobables; d'où on détermine $p_2 = 1/2$.

Dans les autres cas, on établit une récurrence qui est numériquement stable pour calculer p_n .

$$\begin{aligned} p_n \log \frac{1}{p_n} + (1 - p_n) \left(\log \frac{1}{1 - p_n} + \log(n - 1) \right) &= 1 \\ \Rightarrow (1 - p_n) \left(\log \frac{1}{1 - p_n} + \log(n - 1) \right) &= 1 - p_n \log \frac{1}{p_n} \\ \Rightarrow \frac{1 - p_n \log \frac{1}{p_n}}{\log \frac{1}{1 - p_n} + \log(n - 1)} &= 1 - p_n \\ \Rightarrow 1 - \frac{1 - p_n \log \frac{1}{p_n}}{\log \frac{1}{1 - p_n} + \log(n - 1)} &= p_n \end{aligned}$$

Les estimations numériques de p_3 , p_4 et p_5 sont données ici.

$$\begin{aligned} p_3 &\approx 0.7729 \\ p_4 &\approx 0.8107 \\ p_5 &\approx 0.8295 \end{aligned}$$

¹Ici, tous les logarithmes sont calculés à la base 2.

- (b) Soient π_n , π_g et π_b les probabilités stationnaires des états correspondant aux faits d'avoir un pixel voisin qui est noir, gris ou blanc, respectivement. On peut calculer ces probabilités en posant les trois équations évidentes plus celle qui dit que la somme des probabilités stationnaires doit donner 1.

$$\begin{aligned}\pi_n &= 80\% \pi_n + 15\% \pi_g + 5\% \pi_b \\ \pi_g &= 15\% \pi_n + 70\% \pi_g + 15\% \pi_b \\ \pi_b &= 5\% \pi_n + 15\% \pi_g + 80\% \pi_b \\ 1 &= \pi_n + \pi_g + \pi_b\end{aligned}$$

En résolvant ce système d'équations, on obtient que:

$$\pi_n = \pi_g = \pi_b = \frac{1}{3}.$$

L'entropie conditionnelle de l'état 'n' consiste à évaluer l'imprévisibilité du pixel prédit quand son voisin de gauche est un pixel noir.

$$H_n = -80\% \log 80\% - 15\% \log 15\% - 5\% \log 5\% = 0.8842 \text{ bits}$$

On procède similairement pour les deux autres états.

$$\begin{aligned}H_g &= -15\% \log 15\% - 70\% \log 70\% - 15\% \log 15\% = 1.1813 \text{ bits} \\ H_b &= -5\% \log 5\% - 15\% \log 15\% - 80\% \log 80\% = 0.8842 \text{ bits}\end{aligned}$$

Enfin, l'entropie du modèle est la moyenne des entropies conditionnelles pondérée par les probabilités stationnaires des états.

$$H = \frac{1}{3} 0.8842 + \frac{1}{3} 1.1813 + \frac{1}{3} 0.8842 = 0.9832 \text{ bits}$$

3. **Code 1** : il est uniquement décodable. Utilisons l'algorithme de test de décodabilité unique. On commence par initialiser DS. On constate que le mot de code 10 est un préfixe des deux mots de code qui le suivent, menant aux *dangling suffixes* 0 et 00. On a des situations similaires avec les mots de code 110 et 1110, menant aux mêmes *dangling suffixes*. Ainsi, DS est initialisé à $\{0, 00\}$. Lors de la première itération, on compare les mots de code et les *dangling suffixes* pour trouver de nouveaux *dangling suffixes*. En particulier, on voit que les deux *dangling suffixes* 0 et 00 sont des préfixes du mot de code 000. Toutefois, ceci ne mène pas à la découverte de nouveaux *dangling suffixes*. À l'inverse, aucun mot de code n'est préfixe d'un *dangling suffix*. Donc, DS est stabilisé et aucun *dangling suffix* est un mot de code. On conclut que le code est uniquement décodable.

Code 2 : il est uniquement décodable. On procède à peu près de la même manière que pour le code 1. Utilisons l'algorithme de test de décodabilité unique. On commence par initialiser DS. Dans trois cas différents, on a un mot de code préfixe d'un autre, entraînant chaque fois l'identification du *dangling suffix* 10. Ainsi, DS est initialisé à $\{10\}$. Lors de la première itération, on ne découvre rien de plus que le fait que le *dangling suffix* 10 est préfixe du mot de code 1010. Ceci n'ajoute aucun nouveau *dangling suffix* à DS. On conclut que le code est uniquement décodable.

Code 3 : il est uniquement décodable. On procède de la même manière que pour les deux codes précédents. On initialise DS à $\{1000, 1001\}$ et la première itération ne permet pas d'ajouter d'autres *dangling suffixes*. On conclut que le code est uniquement décodable.

Code 4 : il est uniquement décodable car il est un code préfixe.

Code 5 : il n'est pas uniquement décodable car:

$$C(\mathbf{d}) \cdot C(\mathbf{h}) \cdot C(\mathbf{d}) = 001 \cdot 01111 \cdot 001 = 001011 \cdot 11001 = C(\mathbf{e}) \cdot C(\mathbf{k}) .$$

4. (a) Il n'y a pas de tel code préfixe car les longueurs des mots de code ne respectent pas l'inégalité de Kraft-McMillan.

$$\begin{aligned} & \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^3} + \frac{1}{2^5} + \frac{1}{2^4} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^2} \\ = & \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^5} \\ = & \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^4} + \frac{1}{2^4} \\ = & \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^4} \\ = & \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^4} \\ = & \frac{1}{2} + \frac{1}{2} + \frac{1}{2^4} \\ = & 1 + \frac{1}{2^4} \\ > & 1 \end{aligned}$$

- (b) Oui, il y a le code préfixe suivant.

a	111110	d	00
b	11110	i	01
c	100	c	100
d	00	f	101
e	111111	h	110
f	101	g	1110
g	1110	b	11110
h	110	a	111110
i	01	e	111111

ou, présenté autrement,

5. (a) On trie les symboles par fréquences d'apparition. Le code que vous avez construit peut être passablement différent; certains groupes de symboles peuvent être divisés de deux façons différentes à cause d'égalités dans les sommes de fréquences.

```

f : 1  c : 2  l : 4  b : 5  a : 6  k : 7  h : 8  g : 11  i : 12  e : 13  j : 15  d : 16
f : 1  c : 2  l : 4  b : 5  a : 6  k : 7  h : 8  g : 11 | i : 12  e : 13  j : 15  d : 16
f : 1  c : 2  l : 4  b : 5  a : 6  k : 7 | h : 8  g : 11  i : 12  e : 13 | j : 15  d : 16
f : 1  c : 2  l : 4  b : 5 | a : 6  k : 7  h : 8 | g : 11  i : 12 | e : 13  j : 15 | d : 16
f : 1  c : 2 | l : 4  b : 5 | a : 6 | k : 7  h : 8 | g : 11  i : 12 | e : 13 | j : 15 | d : 16
f : 1 | c : 2 | l : 4 | b : 5 | a : 6 | k : 7 | h : 8 | g : 11 | i : 12 | e : 13 | j : 15 | d : 16

```

Donc, le code obtenu est le suivant.

f : 000000		a : 0010
c : 000001		b : 0001
l : 00001		c : 000001
b : 0001		d : 111
a : 0010		e : 101
k : 0011	ou, présenté autrement:	f : 000000
h : 010		g : 011
g : 011		h : 010
i : 100		i : 100
e : 101		j : 110
j : 110		k : 0011
d : 111		l : 00001

$$\begin{aligned}
 L_{\text{moy}} &= \frac{6 * 4 + 5 * 4 + 2 * 6 + 16 * 3 + 13 * 3 + 1 * 6 + 11 * 3 + 8 * 3 + 12 * 3 + 15 * 3 + 7 * 4 + 4 * 5}{100} \\
 &= \frac{(16 + 13 + 11 + 8 + 12 + 15) * 3 + (6 + 5 + 7) * 4 + 4 * 5 + (2 + 1) * 6}{100} \\
 &= \frac{75 * 3 + 18 * 4 + 4 * 5 + 3 * 6}{100} \\
 &= \frac{225 + 72 + 20 + 18}{100} \\
 &= \frac{335}{100}
 \end{aligned}$$

(b) L'algorithme de Huffman doit joindre des symboles pour en faire des frères. Par exemple, à la première étape ci-bas, les symboles f et c sont joints. Nous dénotons par $A \equiv f \swarrow \searrow c$ le fait de joindre ces deux symboles et de les remplacer par un nouveau symbole artificiel nommé A . Cette notation signifie que le mot de code de f sera formé en ajoutant 0 au mot de code de A et que le mot de code de c sera formé en ajoutant 1 au mot de code de A . De plus, la fréquence de A est la somme des fréquences de f et de c .

$f : 1, c : 2, l : 4, b : 5, a : 6, k : 7, h : 8, g : 11, i : 12, e : 13, j : 15, d : 16$

$A \equiv f \swarrow \searrow c$

$A : 3, l : 4, b : 5, a : 6, k : 7, h : 8, g : 11, i : 12, e : 13, j : 15, d : 16$

$B \equiv A \swarrow \searrow l$

$b : 5, a : 6, k : 7, B : 7, h : 8, g : 11, i : 12, e : 13, j : 15, d : 16$

$C \equiv b \swarrow \searrow a$

$k : 7, B : 7, h : 8, g : 11, C : 11, i : 12, e : 13, j : 15, d : 16$

$D \equiv k \swarrow \searrow B$

$h : 8, g : 11, C : 11, i : 12, e : 13, D : 14, j : 15, d : 16$

$E \equiv h \swarrow \searrow g$

$C : 11, i : 12, e : 13, D : 14, j : 15, d : 16, E : 19$

$F \equiv C \swarrow \searrow i$

Donc, le code obtenu est le suivant.

$h : 000$	$a : 0101$	$e : 13, D : 14, j : 15, d : 16, E : 19, F : 23$
$g : 001$	$b : 0100$	$G \equiv e \swarrow \searrow D$
$b : 0100$	$c : 101101$	$j : 15, d : 16, E : 19, F : 23, G : 27$
$a : 0101$	$d : 111$	$H \equiv j \swarrow \searrow d$
$i : 011$	$e : 100$	$E : 19, F : 23, G : 27, H : 31$
$e : 100$	$f : 101100$	$I \equiv E \swarrow \searrow F$
$k : 1010$	$g : 001$	$G : 27, H : 31, I : 42$
$f : 101100$	$h : 000$	$J \equiv G \swarrow \searrow H$
$c : 101101$	$i : 011$	$I : 42, J : 58$
$l : 10111$	$j : 110$	$K \equiv I \swarrow \searrow J$
$j : 110$	$k : 1010$	$K : 100$
$d : 111$	$l : 10111$	

$$\begin{aligned}
 L_{\text{moy}} &= \frac{6 * 4 + 5 * 4 + 2 * 6 + 16 * 3 + 13 * 3 + 1 * 6 + 11 * 3 + 8 * 3 + 12 * 3 + 15 * 3 + 7 * 4 + 4 * 5}{100} \\
 &= \frac{(16 + 13 + 11 + 8 + 12 + 15) * 3 + (6 + 5 + 7) * 4 + 4 * 5 + (2 + 1) * 6}{100} \\
 &= \frac{75 * 3 + 18 * 4 + 4 * 5 + 3 * 6}{100} \\
 &= \frac{225 + 72 + 20 + 18}{100} \\
 &= \frac{335}{100}
 \end{aligned}$$

6. Pour construire le mot de code pour le nombre i , on doit déterminer le numéro du groupe de naturels, $q = \lfloor i/2 \rfloor$, et identifier le membre du groupe, $r = i - 2q \in \{0, 1\}$. Le numéro de groupe est encodé à l'aide du code unaire vu en classe. Le numéro q se traduit en le mot de code $0^q 1$, en $q + 1$ bits. Le numéro de membre est encodé à l'aide d'un code préfixe. Les numéros $r = 0$ et $r = 1$ se traduisent en les mots de code 0 et 1, respectivement.

0 : 10	2 : 010	4 : 0010	6 : 00010
1 : 11	3 : 011	5 : 0011	7 : 00011

Soit $l_2(i)$ la longueur du mot de code pour le naturel i . Nous avons donc que $l_2(i) = q + 2$. Maintenant, soit L_2 la longueur moyenne du mot de code pour un naturel tiré au hasard.

$$\begin{aligned}
L_2 &= \sum_{i=0}^{\infty} p_i \cdot l_2(i) = \sum_{q=0}^{\infty} \sum_{r=0}^1 p_{2q+r} \cdot l_2(2q+r) \\
&= \sum_{q=0}^{\infty} \sum_{r=0}^1 p_{2q+r} \cdot (q+2) \\
&= \sum_{q=0}^{\infty} (q+2) \cdot \sum_{r=0}^1 p_{2q+r} \\
&= \sum_{q=0}^{\infty} (q+2) \cdot (p_{2q} + p_{2q+1}) \\
&= \sum_{q=0}^{\infty} (q+2) \cdot \left(\frac{3}{5} \cdot \left(\frac{1}{2}\right)^{\lfloor 2q/2 \rfloor + 1} + \frac{2}{5} \cdot \left(\frac{1}{2}\right)^{\lfloor (2q+1)/2 \rfloor + 1} \right) \\
&= \sum_{q=0}^{\infty} (q+2) \cdot \left(\frac{3}{5} \cdot \left(\frac{1}{2}\right)^{\lfloor q \rfloor + 1} + \frac{2}{5} \cdot \left(\frac{1}{2}\right)^{\lfloor q+1/2 \rfloor + 1} \right) \\
&= \sum_{q=0}^{\infty} (q+2) \cdot \left(\frac{3}{5} \cdot \left(\frac{1}{2}\right)^{q+1} + \frac{2}{5} \cdot \left(\frac{1}{2}\right)^{q+1} \right) \\
&= \sum_{q=0}^{\infty} (q+2) \cdot \left(\frac{1}{2}\right)^{q+1} = \sum_{q=0}^{\infty} \frac{1}{2} \cdot (q+2) \cdot \left(\frac{1}{2}\right)^q \\
&= \sum_{q=0}^{\infty} \frac{q}{2} \cdot \left(\frac{1}{2}\right)^q + \sum_{q=0}^{\infty} \left(\frac{1}{2}\right)^q \\
&= \frac{1}{2} \cdot \sum_{q=0}^{\infty} q \cdot \left(\frac{1}{2}\right)^q + \sum_{q=0}^{\infty} \left(\frac{1}{2}\right)^q \\
&= \frac{1}{2} \cdot \frac{\frac{1}{2}}{\left(1 - \frac{1}{2}\right)^2} + \frac{1}{1 - \frac{1}{2}} = \frac{1}{2} \cdot \frac{\frac{1}{2}}{\left(\frac{1}{2}\right)^2} + \frac{1}{\frac{1}{2}} \\
&= \frac{1}{2} \cdot \frac{1}{\frac{1}{2}} + 2 = 1 + 2 = 3
\end{aligned}$$

7. On fait croître le dictionnaire de mots sources tant qu'on reste à l'intérieur de la limite de 12 entrées.

a	3/10
b	2/10
c	5/10

a	3/10
b	2/10
ca	15/100
cb	10/100
cc	25/100

aa	9/100
ab	6/100
ac	15/100
b	2/10
ca	15/100
cb	10/100
cc	25/100

aa	9/100
ab	6/100
ac	15/100
b	2/10
ca	15/100
cb	10/100
cca	75/1000
ccb	50/1000
ccc	125/1000

aa	9/100
ab	6/100
ac	15/100
ba	6/100
bb	4/100
bc	10/100
ca	15/100
cb	10/100
cca	75/1000
ccb	50/1000
ccc	125/1000

8. Le tableau suivant indique l'état de l'intervalle avant et après le traitement de chaque caractère de l'entrée à coder. L'état du compteur $E_3.count$ est aussi indiqué. Enfin, l'intervalle résultant du codage du symbole est redimensionné si nécessaire. Les éventuels bits émis par le redimensionnement sont indiqués.

# SYMBOLE	I	II	III	IV
INT./ $E_3.count$ DÉBUT	$[\frac{0}{10}, \frac{10}{10})/0$	$[\frac{0}{10}, \frac{10}{10})/0$	$[\frac{0}{100}, \frac{60}{100})/0$	$[\frac{10}{100}, \frac{70}{100})/1$
INT. POUR a	$[\frac{0}{10}, \frac{3}{10})$	$[\frac{0}{10}, \frac{3}{10})$	$[\frac{0}{100}, \frac{18}{100})$	$[\frac{10}{100}, \frac{28}{100})$
INT. POUR b	$[\frac{3}{10}, \frac{5}{10})$	$[\frac{3}{10}, \frac{5}{10})$	$[\frac{18}{100}, \frac{30}{100})$	$[\frac{28}{100}, \frac{40}{100})$
INT. POUR c	$[\frac{5}{10}, \frac{10}{10})$	$[\frac{5}{10}, \frac{10}{10})$	$[\frac{30}{100}, \frac{60}{100})$	$[\frac{40}{100}, \frac{70}{100})$
INT./ $E_3.count$ MILIEU	$[\frac{5}{10}, \frac{10}{10})/0$	$[\frac{0}{10}, \frac{3}{10})/0$	$[\frac{30}{100}, \frac{60}{100})/0$	$[\frac{28}{100}, \frac{40}{100})/1$
REDIM. & BITS	$E_2 : 1$	$E_1 : 0$	$E_3 : \epsilon$	$E_1 : 01$
INT./ $E_3.count$	$[\frac{0}{10}, \frac{10}{10})/0$	$[\frac{0}{10}, \frac{6}{10})/0$	$[\frac{10}{100}, \frac{70}{100})/1$	$[\frac{56}{100}, \frac{80}{100})/0$
REDIM. & BITS				$E_2 : 1$
INT./ $E_3.count$				$[\frac{12}{100}, \frac{60}{100})/0$
REDIM. & BITS				
INT./ $E_3.count$				
INT./ $E_3.count$ FIN	$[\frac{0}{10}, \frac{10}{10})/0$	$[\frac{0}{10}, \frac{6}{10})/0$	$[\frac{10}{100}, \frac{70}{100})/1$	$[\frac{12}{100}, \frac{60}{100})/0$

Cette fois-ci, l'intervalle est géré en nombres entiers. L'usage d'aussi peu que 16 tuiles mène à de sévères distorsions des probabilités.

# SYMBOLE	I	II	III	IV
INT./ $E_3.count$ DÉBUT	$[0, 15]/0$	$[0, 15]/0$	$[0, 15]/0$	$[0, 15]/0$
INT. POUR a	$[0, 3]$	$[0, 3]$	$[0, 3]$	$[0, 3]$
INT. POUR b	$[4, 7]$	$[4, 7]$	$[4, 7]$	$[4, 7]$
INT. POUR c	$[8, 15]$	$[8, 15]$	$[8, 15]$	$[8, 15]$
INT./ $E_3.count$ MILIEU	$[8, 15]/0$	$[0, 3]/0$	$[8, 15]/0$	$[4, 7]/0$
REDIM. & BITS	$E_2 : 1$	$E_1 : 0$	$E_2 : 1$	$E_1 : 0$
INT./ $E_3.count$	$[0, 15]/0$	$[0, 7]/0$	$[0, 15]/0$	$[8, 15]/0$
REDIM. & BITS		$E_1 : 0$		$E_2 : 1$
INT./ $E_3.count$		$[0, 15]/0$		$[0, 15]/0$
REDIM. & BITS				
INT./ $E_3.count$				
INT./ $E_3.count$ FIN	$[0, 15]/0$	$[0, 15]/0$	$[0, 15]/0$	$[0, 15]/0$