

Problème du Serpent
Dénombrement de chaînes dans un graphe rectangle

Théo RUDKIEWICZ

13 juin 2020

Sommaire

1	Introduction	3
1.1	Définitions	3
2	Cas d'un graphe rectangle quelconque	5
2.1	Parité	5
2.2	Théorème réducteur	6
3	Graphe $n \times 1$	9
4	Graphe $n \times 2$	10
4.1	Nombre de parcours à partir d'un angle	10
4.2	Parcours depuis un côté	11
4.3	Conclusion	11
5	Graphe $n \times 3$	12
5.1	$P_B(3 \times n)$	12
5.2	$P_B(n \times 3); P_O(n \times 3)$	13
5.3	$P_A(n \times 3)$	14
5.4	Trajet $(n \times 3; (1, 2); (\lambda; \lambda))$	18
5.5	Trajets depuis les milieux : Trajet $(n \times 3; (\lambda, 2); (\lambda; \lambda))$	18
5.6	Trajets depuis les côtés : Trajet $(n \times 3; (\lambda, 1); (\lambda; \lambda))$	22
6	Algorithme en Python	25

Chapitre 1

Introduction

On cherche à dénombrer l'ensemble des chaînes hamiltoniennes orientées dans un graphe rectangle de taille $L \times l$. (cf Fig 1.1)

En utilisant des listes, structure plus adaptée à une implémentation, on peut définir le problème comme suit. Un parcours est une liste ordonnée de positions. Soit P un parcours. $L \times l$ est la taille du graphe.

$$P \text{ est valide} \iff \begin{cases} (1) \forall (x, y) \in \llbracket 1; L \rrbracket \times \llbracket 1; l \rrbracket; (x, y) \in P & (\text{Tout les points sont atteints}) \\ (2) \forall p \in P; p \in \llbracket 1; L \rrbracket \times \llbracket 1; l \rrbracket & (\text{Rien ne dépasse du carré}) \\ (3) \forall (p, q) \in P^2; p \neq q & (\text{Aucun point n'est atteint deux fois}) \\ (4) \forall i \in \llbracket 1; L \times l \rrbracket / P_i = (x, y); [P_{i+1} = (x \pm 1, y)] \vee [P_{i+1} = (x, y \pm 1)] & (\text{Tout les déplacements sont justes}) \end{cases}$$

On cherche $\max(\text{card}(\Omega))$

$$\Omega = \bigcup P / P \text{ est valide}$$

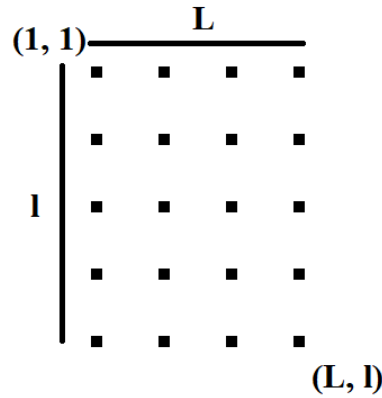


FIGURE 1.1 – Représentation du graphe 4×5

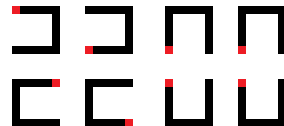


FIGURE 1.2 – Représentation de Ω pour $G(2 \times 2)$

1.1 Définitions

- On note $G(L \times l)$ le graphe rectangle de taille L par l . Par convention, on note G' les sous-graphes.

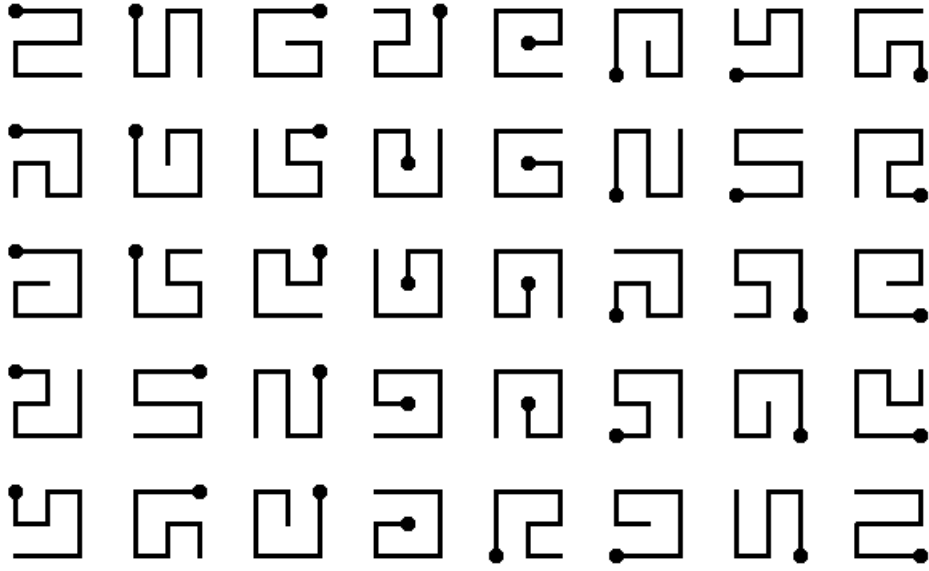


FIGURE 1.3 – Représentation de Ω pour $G(3 \times 3)$

- On attribue des coordonnées à chaque point. La première est l'avancement sur la longueur L (par convention horizontale vers la droite). La deuxième est l'avancement sur la largeur l (par convention verticale vers le bas). Ainsi les coordonnées vont de $(1, 1)$ à (L, l) .
- On utilise par la suite certaines dénominations pour des motifs précis :
 - en colimaçon \sqsupset
 - en direct \llcorner

$$\text{Trajet} \left| \begin{array}{ccc} (\mathbb{N}^{*2})^3 & \longrightarrow & \mathbb{N} \\ (L \times l); (x, y); (\chi, v) & \longmapsto & \text{Nombre de parcours de } (x, y) \text{ vers } (\chi, v) \text{ dans le graphe } G(L \times l) \end{array} \right.$$

$$\text{Trajet}(L \times l; (x, y); (\lambda; \lambda)) = \text{Trajet}(L \times l; (\lambda, \lambda); (x, y)) = \sum_{\chi=1; v=1}^{\chi=L; v=l} \text{Trajet}(L \times l; (x, y); (\chi; v))$$

$$\text{Parcours}(L \times l) = \sum_{x=1; y=1}^{x=L; y=l} \text{Trajet}(L \times l; (x, y); (\lambda; \lambda))$$

- $P_A(L \times l) = \text{Trajet}(L \times l; (1, 1); (\lambda; \lambda))$
- $P_B(L \times l) = \text{Trajet}(L \times l; (1, 1); (L; 1)) \neq P_B(l \times L) = \text{Trajet}(L \times l; (1, 1); (1; l))$
- $P_O(L \times l) = \text{Trajet}(L \times l; (1, 1); (L; l))$
- Une case de coordonnées (x, y) est dite paire si et seulement si $x + y$ est paire et impaire si et seulement si $x + y$ est impaire.

$$\text{Parité} \left| \begin{array}{ccc} \mathbb{N}^2 & \longrightarrow & \{0; 1\} \\ (x, y) & \longmapsto & (x + y) \% 2 \end{array} \right.$$

- Un chemin, contrairement à un parcours ne prend pas en compte le sens dans lequel le serpent est mis. Ainsi : $[(0, 0); (1, 0)] \Leftrightarrow [(1, 0); (0, 0)]$. Il s'agit alors d'une chaîne hamiltonienne dans un graphe non orienté.

Chapitre 2

Cas d'un graphe rectangle quelconque

2.1 Parité

2.1.1 Lemme : répartition de la parité

Lemme.

$$2 \nmid L \times l \Leftrightarrow \begin{cases} \text{nombre de cases paires} = \frac{L \times l + 1}{2} \\ \text{nombre de cases impaires} = \frac{L \times l - 1}{2} \end{cases} \quad \square$$

$$2 \mid L \times l \Leftrightarrow \text{nombre de cases paires} = \text{nombre de cases impaires} = \frac{L \times l}{2} \quad \square$$

2.1.2 Théorème parité

Théorème 2.1 (Théorème parité).

1.

$$\forall i \in \llbracket 1; L \times l \rrbracket; \text{Parité}(P_i) \neq \text{Parité}(P_{i+1})$$

2.

$$\begin{cases} 2 \mid L \times l \Leftrightarrow \text{Parité}(P_0) \neq \text{Parité}(P_{-1}) \\ 2 \nmid L \times l \Leftrightarrow \text{Parité}(P_0) = \text{Parité}(P_{-1}) \end{cases}$$

1. *Démonstration.* D'après la définition d'un parcours valide :

$$(4) \forall i \in \llbracket 1; L \times l \rrbracket / P_i = (x, y); [P_{i+1} = (x \pm 1, y)] \vee [P_{i+1} = (x, y \pm 1)] \text{ (Tout les déplacements sont justes)}$$

Or $x + y \neq (x + y) \pm 1[2]$ donc :

$$\forall i \in \llbracket 1; L \times l \rrbracket; \text{Parité}(P_i) \neq \text{Parité}(P_{i+1})$$

■

2. *Démonstration.* Dans $G(L \times l)$, il y a $L \times l - 1$ arêtes empruntées, or d'après la propriété (1), cela correspond à $L \times l - 1$ changement de parité. Les changements de parités s'annulent deux à deux donc seul la parité du nombre de changements compte. Ainsi si le nombre de changements est pair, la parité est conservée sinon elle est inversée. D'où :

$$\begin{cases} 2 \nmid L \times l - 1 \Leftrightarrow 2 \mid L \times l \Leftrightarrow \text{Parité}(P_0) \neq \text{Parité}(P_{-1}) \\ 2 \mid L \times l - 1 \Leftrightarrow 2 \nmid L \times l \Leftrightarrow \text{Parité}(P_0) = \text{Parité}(P_{-1}) \end{cases}$$

■

2.1.3 Application du théorème parité aux graphes impairs

Corollaire 2.1.1 (Départs impossibles).

$$2 \nmid L \times l \Leftrightarrow \neg \exists P / (P \text{ est valide} \wedge \text{Parité}(P_0) = 1)$$

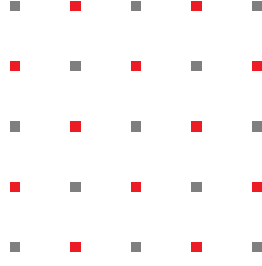


FIGURE 2.1 – Application du corollaire 2.1.1 au graphe carré 5×5 (avec en rouge les cases d'où aucun parcours ne part)

Démonstration. Par l'absurde : On suppose qu'il existe un parcours partant d'une case impaire allant vers une case impaire. Entre ces deux points il devra parcourir $\frac{L \times l + 1}{2}$ case paires et $\frac{L \times l - 1}{2} - 2$ case impaires. Or le parcours est forcément constitué d'une alternance de cases paires et impaires. Ce trajet n'existe donc pas. ■

2.2 Théorème réducteur

Théorème 2.2 (Théorème Réducteur). 1. *Réduction simple :*

$$\forall (L; l) \in \mathbb{N}^{*2}; P_A(L \times l) \geq P_A(L \times l - 1) + P_A(L - 1 \times l)$$

2. *Réduction en colimaçon impairs :*

$$(a) \forall (L; l) \in \mathbb{N}^{*2} / (2 \nmid L);$$

$$P_A(L \times l) \geq \text{Red}(l) + \sum_{h=1}^{l-1} P_A(L \times h)$$

$$(b) \forall (L; l) \in \mathbb{N}^{*2} / (2 \nmid l);$$

$$P_A(L \times l) \geq \text{Red}(L) + \sum_{h=1}^{L-1} P_A(h \times l)$$

3. *Réduction en colimaçon pairs :*

$$(a) \forall (L; l) \in \mathbb{N}^{*2} / (2 \mid L);$$

$$P_A(L \times l) \geq \text{Red}(l) + P_A(L \times l - 1) + \sum_{h=1}^{l-2} \sum_{x=1; 2 \nmid x}^{L-1} \text{Trajet}(L \times h; (1, 1); (x; 1))$$

$$(b) \forall (L; l) \in \mathbb{N}^{*2} / (2 \mid l);$$

$$P_A(L \times l) \geq \text{Red}(L) + P_A(L - 1 \times l) + \sum_{h=1}^{L-2} \sum_{x=1; 2 \nmid x}^{l-1} \text{Trajet}(h \times l; (1, 1); (1; x))$$

4. *Réduction combinées :*

$$(a) \forall (L; l) \in (2\mathbb{N}^*)^2; P_A(L \times l) \geq$$

$$\sum_{h=1}^{L-1} P_A(h \times l) + \sum_{h=1}^{l-1} P_A(L \times h)$$

$$(b) \forall (L; l) \in \mathbb{N}^{*2}/(2 \mid L \wedge 2 \mid l); P_A(L \times l) \geq$$

$$P_A(L \times l - 1) + \sum_{h=1}^{l-2} \sum_{x=1; 2 \nmid x}^{L-1} \text{Trajet}(L \times h; (1, 1); (x; 1)) + P_A(L - 1 \times l) + \sum_{h=1}^{L-2} \sum_{x=1; 2 \nmid x}^{l-1} \text{Trajet}(h \times l; (1, 1); (1; x))$$

$$(c) \forall (L; l) \in \mathbb{N}^{*2}/(2 \nmid L \wedge 2 \mid l); P_A(L \times l) \geq$$

$$\sum_{h=1}^{l-1} P_A(L \times h) + P_A(L - 1 \times l) + \sum_{h=1}^{L-2} \sum_{x=1; 2 \nmid x}^{l-1} \text{Trajet}(h \times l; (1, 1); (1; x))$$

$$(d) \forall (L; l) \in (2\mathbb{N}^* + 1)^2; P_A(L \times l) \geq$$

$$P_A(L \times l - 1) + \sum_{h=1}^{l-2} \sum_{x=1; 2 \nmid x}^{L-1} \text{Trajet}(L \times h; (1, 1); (x; 1)) + \sum_{h=1}^{l-1} P_A(L \times h)$$

Démonstration. 1. On cherche à montrer que l'on peut se ramener au parcours d'un sous-graphe. On part ainsi d'un graphe $G(L \times l)$ et on le réduit à un graphe $G'(L \times l - 1)$. On peut effectuer cette réduction en longeant simplement le bord (cf Fig. 2.2). Ainsi on peut effectuer cette réduction des deux cotés. On a alors à parcourir $G'(L \times l - 1)$ et $G'(L - 1 \times l)$. De plus, ces deux réductions sont compatibles donc elles ne compteront pas deux fois la même chose. D'où :

$$\forall (L; l) \in \mathbb{N}^{*2}; P_A(L \times l) \geq P_A(L \times l - 1) + P_A(L - 1 \times l) \quad \blacksquare$$

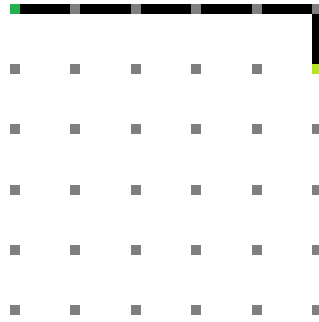


FIGURE 2.2 – Réduction au parcours du sous-graphe $G'(L \times l - 1)$ (en vert foncé le départ et en vert clair le nouveau départ)

2. On cherche à montrer l'on peut se ramener au parcours d'un ensemble de sous-graphe. L est impair donc on peut réduire par colimaçon (cf Fig. 2.3). Pour réduire aux sous-graphes $G'(L \times l - 2)$, $G'(L \times l - 3)$, ..., $G'(L \times 1)$. On peut également effectuer cette réduction à partir du côté l . De plus, ces réductions sont compatibles avec la réduction simple et compatibles entre elles. D'où :

$$\begin{cases} \forall (L; l) \in \mathbb{N}^{*2}/(2 \nmid L); P_A(L \times l) \geq \text{Red}(l) + \sum_{h=1}^{l-1} P_A(L \times h) \\ \forall (L; l) \in \mathbb{N}^{*2}/(2 \nmid l); P_A(L \times l) \geq \text{Red}(L) + \sum_{h=1}^{L-1} P_A(h \times l) \end{cases} \quad \blacksquare$$

FIGURE 2.3 – Réduction au parcours du sous-graphe $G'(L \times l - 3)$ (en vert foncé le départ et en vert clair le nouveau départ)

FIGURE 2.4 – Réduction au parcours du sous-graphe $G'(L \times l - 3)$ avec arrivée contrainte (en vert foncé le départ et en vert clair le nouveau départ, en rouge le départ et en orange la nouvelle arrivée)

3. Dans le cas d'une longueur impaire, une réduction par colimaçon laisse forcément un trou (cf Fig. 2.4). Ce trou doit forcément être le point d'arrivée car il y est impossible de faire demi-tour. De plus ce trou ne peut être présent à n'importe quel point : il sera forcément impair pour satisfaire le théorème de parité (cf. 2.1). Ainsi le départ et l'arrivée du parcours sont fixés. D'où :

$$\begin{cases} \forall (L; l) \in \mathbb{N}^{*2}/(2 \mid L); P_A(L \times l) \geq Red(l) + P_A(L \times l - 1) + \sum_{h=1}^{l-2} \sum_{x=1; 2 \nmid x}^{L-1} \text{Trajet}(L \times h; (1, 1); (x, 1)) \\ \forall (L; l) \in \mathbb{N}^{*2}/(2 \mid l); P_A(L \times l) \geq Red(L) + P_A(L - 1 \times l) + \sum_{h=1}^{L-2} \sum_{x=1; 2 \nmid x}^{l-1} \text{Trajet}(h \times l; (1, 1); (1, x)) \end{cases}$$

4. Les réductions paires sur un côté et impaires de l'autre ne peuvent aboutir au même trajet. Par exemple, si l'on réduit de façon impaire de 3 la longueur L et de 5 la largeur l aucun des parcours de la première réduction ne seront possibles après la seconde. ■

Chapitre 3

Graphe $n \times 1$

Proposition 3.1.

$$\forall n \in \mathbb{N}^*; P_A(1 \times n) = 1 \quad \square \quad (3.1)$$

$$\forall n \in \mathbb{N}^*; P_O(1 \times n) = 1 \quad \square \quad (3.2)$$

$$\forall n \in \mathbb{N}^* \setminus \{1\}; P_B(1 \times n) = 0 \quad \square \quad (3.3)$$

$$\forall n \in \mathbb{N}^*; P_B(n \times 1) = 1 \quad \square \quad (3.4)$$

$$\forall n \in \mathbb{N}^*; Parcours(1 \times n) = 2 \quad \square \quad (3.5)$$

Chapitre 4

Graphe $n \times 2$

4.1 Nombre de parcours à partir d'un angle

Proposition 4.1 ($P_A(n \times 2)$). *Quel que soit un point atteignable (d'après le théorème parité) à partir de l'angle, il existe un unique parcours qui arrive à ce point.*

$$\forall n \in \mathbb{N}^*; P_A(n \times 2) = n$$

Proposition 4.2. • $P_B(2 \times L) = 1$

- Si $2 \mid n : P_B(L \times 2) = 1$
 Si $2 \nmid n : P_B(L \times 2) = 0$
- Si $2 \mid n : P_O(L \times 2) = 0$
 Si $2 \nmid n : P_O(L \times 2) = 1$

Démonstration de la propriété 4.1. Au premier point il y a deux situations possibles :

- partir vers la droite et donc aller jusqu'à l'autre extrémité pour pouvoir atteindre la case sur le bord opposé de la case de départ, c'est un parcours direct, on atteint alors le seul point atteignable d'après le théorème parité. (cf Fig. 4.1)



FIGURE 4.1 – Parcours direct en U

- faire un motif d'escalier et se retrouver dans exactement la même situation avec un sous-graphe $G'(n - 1 \times 2)$ à parcourir. On ne peut alors plus atteindre le point non-atteignable d'après le théorème de parité de la deuxième ligne. (cf Fig. 4.2)



FIGURE 4.2 – Réduction au parcours du sous-graphe $G'(n - 1 \times 2)$ (en vert foncé le départ et en vert clair le nouveau départ)

On a alors que $P_A(n + 1 \times 2) = 1 + P_A(n \times 2)$. Il s'agit ici de l'application stricte du théorème réducteur simple. (cf 2.2). Par ailleurs dans le cas où $n = 1$, il y a un seul trajet à partir d'une extrémité donc $P_A(2 \times 1) = 1$. Par récurrence ; $\forall n \in \mathbb{N}^*; P_A(2 \times n) = n$. De plus, on remarque que tous les points que le théorème parité ne rend pas inatteignable sont atteints une unique fois (ce qui est confirmé par le nombre de point atteignables $\frac{n \times 2}{2} = n = P_A(2 \times n)$). ■

Démonstration de la propriété 4.2. Le graphe est pair : $2 \mid L \times 2 \Leftrightarrow \text{Parité}(P_0) \neq \text{Parité}(P_{-1})$.
Le point à atteindre pour $P_B(2 \times L)$ est de coordonnée $(1, 2)$ donc de parité impaire, d'où :

$$P_B(2 \times L) = 1$$

- Le point à atteindre pour $P_B(L \times 2)$ est de coordonnée $(L, 1)$ donc de parité inverse à L , d'où : Si $2 \mid n : P_B(L \times 2) = 1$
Si $2 \nmid n : P_B(L \times 2) = 0$
- Le point à atteindre pour $P_O(L \times 2)$ est de coordonnée $(L, 2)$ donc de même parité que L , d'où : Si $2 \mid n : P_O(L \times 2) = 0$
Si $2 \nmid n : P_O(L \times 2) = 1$

■

4.2 Parcours depuis un côté

Proposition 4.3 (Parcours depuis un côté).

$$\forall h \in \mathbb{N}^* \setminus \{1; n\}; \text{Trajet}(n \times 2; (h, 1); (\lambda; \lambda)) = \text{Trajet}(n \times 2; (h, 2); (\lambda; \lambda)) = n - 1$$

Démonstration. A partir d'un point lambda, on est obligé de partir parcourir soit la partie au dessus soit celle du dessous. De plus ce parcours doit s'effectuer en direct pour ne pas couper le graphe en deux parties incomplètes. Lors qu'on revient un cran plus loin que le point de départ, on arrive alors sur l'extrémité d'un sous-graphe à parcourir. (cf Fig.4.3)



FIGURE 4.3 – Réduction au parcours de deux nouveaux sous-graphes : $G'(n - h \times 2)$ et de $G'(h - 1 \times 2)$ (en vert foncé le départ et en vert clair le nouveau départ)

Par conséquent le nombre de parcours pour un point est la somme du nombre de parcours dans les deux sous-graphes $G'(n - h \times 2)$ et $G'(h - 1 \times 2)$, ces parcours s'effectuent à partir d'une extrémité d'où :

$$\text{Trajet}(n \times 2; (h, 1); (\lambda; \lambda)) = P_A(n - h \times 2) + P_A(h - 1 \times 2) = n - h + h - 1 = n - 1$$

■

4.3 Conclusion

Théorème 4.4 (Parcours($n \times 2$)).

$$\forall n \in \mathbb{N}^* \setminus \{1\}; \text{Parcours}(n \times 2) = 2(n^2 - n + 2)$$

Démonstration. Pour $n > 1$, il y a 4 angles et 2 côtés sur chaque côté il y a $n - 2$ points qui ne sont pas des angles. On a alors :

$$\forall n \in \mathbb{N}^* \setminus \{1\}; \text{Parcours}(n \times 2) = 4 \times P_A(n \times 2) + 2(n - 2) \times \text{Trajet}(n \times 2; (h, 1); (\lambda; \lambda)) \quad (4.1)$$

$$= 4n + 2(n - 2)(n - 1) \quad (4.2)$$

$$= 2(n^2 - n + 2) \quad (4.3)$$

■

Chapitre 5

Graphe $n \times 3$

5.1 $P_B(3 \times n)$

Proposition 5.1 ($P_B(3 \times n)$).

1. Si $2 \mid n$:

$$P_B(3 \times n) = 0$$

2. Sinon alors $2 \nmid n$:

$$P_B(3 \times n) = 2^{\frac{n-1}{2}}$$

Démonstration. 1.

$$2 \mid n \Leftrightarrow 2 \mid 3n \Leftrightarrow \text{Parité}(P_0) \neq \text{Parité}(P_{-1})$$

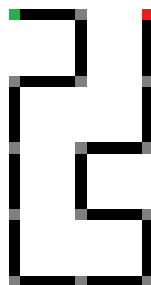
Or on cherche à aller de $(1; 1)$ à $(1; 3)$. Or $\text{Parité}((1, 1)) = \text{Parité}((3, 3))$ donc d'après le théorème parité (cf 2.1), c'est impossible :

$$P_B(3 \times n) = 0$$

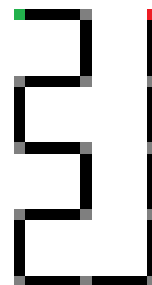
2. $2 \nmid n$ donc on peut faire un parcours en colimaçon le long d'un côté ou de l'autre en descendant puis remontant. On montre alors qu'il y a exactement $\frac{n-1}{2}$ motifs à répartir d'un côté ou de l'autre (cf Fig. 5.1) On utilise les méthodes de dénombrement classique et on obtient :

$$P_B(3 \times n) = 2^{\frac{n-1}{2}}$$

■



(a) Version
(Gauche,
Droite)



(b) Version
(Gauche,
Gauche)

FIGURE 5.1 – Deux parcours pour $P_B(3 \times n)$

5.2 $P_B(n \times 3); P_O(n \times 3)$

Propriétés

1. Lemme arithmétique :

$$1 + \sum_{k=0}^{n-1} 2^k = 2^n \quad \square$$

- 2.

$$P_B(1 \times 3) = 0; P_O(1 \times 3) = 1 \quad \square$$

- 3.

$$\forall n \in \mathbb{N}^*; P_B(n \times 3) = \sum_{h=1}^{n-1} P_O(h \times 3)$$

- 4.

$$\forall n \in \mathbb{N}^*; P_O(n \times 3) = 1 + \sum_{h=1}^{n-1} P_B(h \times 3)$$

- 5.

$$\forall n \in \mathbb{N}^* \setminus \{1\}; P_B(n \times 3) = P_O(n \times 3) = 2^{n-2}$$

Démonstrations

3. Pour aller de $(1; 1)$ vers $(L; 1)$ on peut soit partir vers la droite, puis au bout de h ($h \in \llbracket 1; n-1 \rrbracket$: si on part tout de suite $h = 1$ et si $h = n$ on a déjà atteint la destination) sommets descendre puis revenir sur la gauche puis redescendre laissant ainsi un sous-graphe $G'(3 \times n - h)$ à parcourir d'un angle à son opposé (cf Fig. 5.2). D'où :

$$\forall n \in \mathbb{N}^*; P_B(n \times 3) = \sum_{h=n-(n-1)}^{n-1} P_O(h \times 3) = \sum_{h=1}^{n-1} P_O(h \times 3) \quad \blacksquare$$



FIGURE 5.2 – Reduction à $P_O(n - 3 \times 3)$

4. On peut appliquer le même principe que pour aller de $(1; 1)$ vers $(L; l)$ (cf Fig. 5.3). Seulement on doit alors parcourir le sous-graphe d'un bord à l'autre. De plus le cas $h = 0$ n'est pas exclus, cela rajoute donc un cas. D'où :

$$\forall n \in \mathbb{N}^*; P_O(n \times 3) = 1 + \sum_{h=1}^{n-1} P_B(h \times 3) \quad \blacksquare$$



FIGURE 5.3 – Reduction à $P_B(n - 3 \times 3)$

5. •

$$\forall n \in \mathbb{N}^* \setminus \{1\}; \mathcal{P}_n \text{''} P_B(n \times 3) = P_O(n \times 3) = 2^{n-2},$$

• $n = 2$

$$\begin{cases} P_B(2 \times 3) = 1 \\ P_O(2 \times 3) = 1 \\ 2^{2-2} = 1 \end{cases} \Rightarrow \mathcal{P}_2 \quad \square$$

• On suppose que : $\exists n / \forall i \in \llbracket 2; n-1 \rrbracket; \mathcal{P}_i$

$$P_B(n \times 3) = \sum_{h=1}^{n-1} P_O(h \times 3) = P_O(1 \times 3) + \sum_{h=2}^{n-1} P_O(h \times 3) = 1 + \sum_{h=2}^{n-1} 2^{h-2}$$

$$P_O(n \times 3) = 1 + \sum_{h=1}^{n-1} P_B(h \times 3) = 1 + P_B(1 \times 3) + \sum_{h=2}^{n-1} P_O(h \times 3) = 1 + \sum_{h=2}^{n-1} 2^{h-2}$$

$$1 + \sum_{h=2}^{n-1} 2^{h-2} = 1 + \sum_{h=0}^{(n-1)-2} 2^h = 2^{n-2} \quad \square$$

• $\begin{cases} \mathcal{P}_0 \\ \mathcal{P}_n \Rightarrow \mathcal{P}_{n+1} \end{cases} \Rightarrow \forall n \in \mathbb{N}^* \setminus \{1\}; P_B(n \times 3) = P_O(n \times 3) = 2^{n-2} \quad \blacksquare$

5.3 $P_A(n \times 3)$

On considère par convention et dans toute la suite que $P_A(3 \times 0) = 1$.

Propriétés

1.

$$P_A(n \times 3) = \sum_{h=1}^{n-1} P_A(h \times 3) + n + R(n)$$

2.

$$R(n) = \sum_{h=(n+1)\%2; h+=2}^{n-3} P_A(h \times 2) \times P_B(3 \times n - h - 2)$$

3. (a) Si $2 \mid n$:

$$R(n) = 3 \times 2^{\frac{n-2}{2}} - n - 1$$

(b) Si $2 \nmid n$:

$$R(n) = 5 \times 2^{\frac{n-3}{2}} - n - 1$$

4. D'après l'OEIS¹ :

$$\forall n \in \mathbb{N}^* \setminus \{1\}; P_A(n \times 3) = -2^{-\frac{1}{4}(-1)^n + \frac{1}{2}n - \frac{7}{4}}((-1)^n + 4) + 11 \cdot 2^{n-3}$$

1. The On-Line Encyclopedia of Integer Sequences

Démonstrations

1. On applique simplement le théorème réducteur en colimaçon du côté $l = 3$ et le théorème réducteur simple du côté $L = n$. On a alors :

$$P_A(n \times 3) \geq \sum_{h=1}^{n-1} P_A(h \times 3) + P_A(n \times 2)$$

Le théorème propose seulement une minoration, on note $R(n)$ ce qui n'est pas compté.

$$P_A(n \times 3) = \sum_{h=1}^{n-1} P_A(h \times 3) + P_A(n \times 2) + R(n) = \sum_{h=1}^{n-1} P_A(h \times 3) + n + R(n) \quad \blacksquare$$

2. A partir de $(1; 1)$ on peut partir à droite h fois ($h \in \llbracket 0; n-3 \rrbracket$ car h et le nombre d'arêtes et il y en a $n-1$, $h = n-1$ c'est le cas de la réduction simple, $h = n-2$ on laisse l'angle non-parcouru). Après h déplacements on part vers le bas une seule fois pour ne pas couper le graphe en deux parties non pleines. Puis on peut partir vers la gauche et alors revenir jusqu'à l'extrémité mais dans ce cas là il s'agit de la réduction classique évoquée dans 1. Il faut donc partir à droite puis remonter pour ne pas isoler un point. Comme ce motif en colimaçon délimite un passage de 1 qui ne permet pas de faire demi-tour, l'arrivée sera à gauche de ce premier motif. On se retrouve avec deux zones à parcourir (cf Fig. 5.4). La première doit revenir deux cases en dessous pour finir par la zone à gauche du motif en colimaçon. La zone à droite est le sous-graphe $G'(n-h-2 \times 3)$ et celui à gauche $G'(h \times 2)$. Or nous savons que (cf 5.1) :

$$P_B(3 \times n - h - 2) \neq 0 \Leftrightarrow 2 \nmid n - h - 2 \Leftrightarrow 2 \nmid n - h \Leftrightarrow n \not\equiv h[2]$$

Ainsi on commence donc avec $h = (n+1)\%2$ puis on va de 2 en 2 pour satisfaire ce critère. On a donc à faire la somme du nombre de parcours possibles pour tous les h possibles. Ce nombre de parcours est d'après le lemme des bergers $P_A(h \times 2) \times P_B(3 \times n - h - 2)$. D'où :

On considère ici que $P_A(3 \times 0) = 1$.

$$R(n) = \sum_{h=(n+1)\%2; h+=2}^{n-3} P_A(h \times 2) \times P_B(3 \times n - h - 2) \quad \blacksquare$$

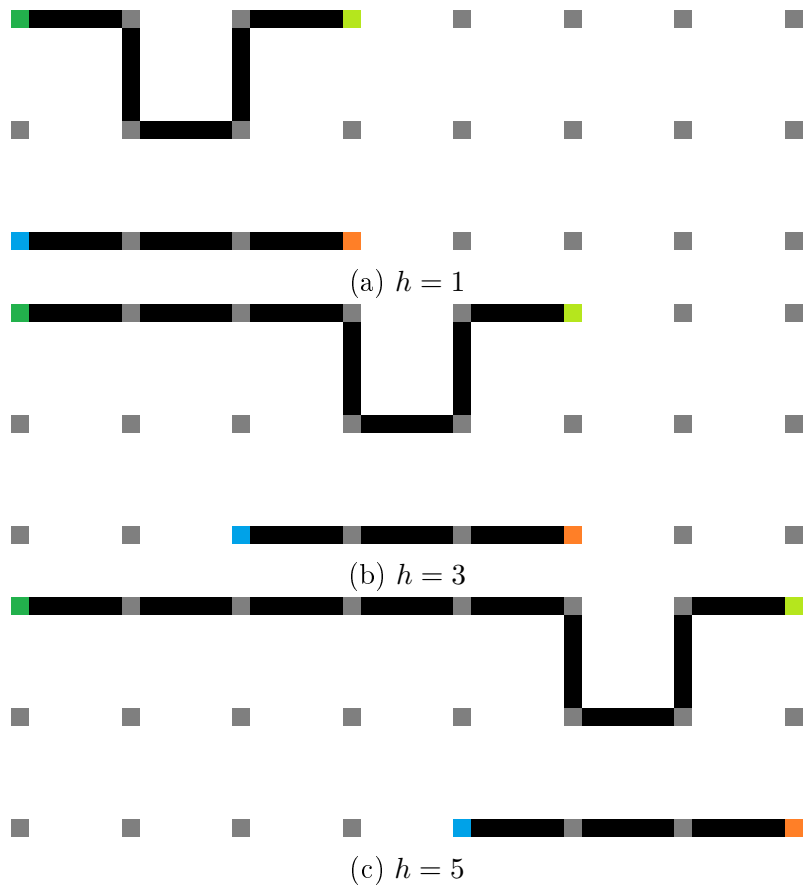


FIGURE 5.4 – Parcours restants après réduction par théorème réducteur (en vert foncé le départ, en vert clair le départ du premier sous-graphe de droite, en orange l'arrivée du sous-graphe de droite et en cyan le départ du sous-graphe de gauche)

3. (a) Si $2 \mid n$:

$$R(n) = \sum_{h=1; h+=2}^{n-3} P_A(h \times 2) \times P_B(3 \times n - h - 2) \quad (5.1)$$

$$= \sum_{h=1; h+=2}^{n-3} h \times 2^{\frac{(n-h-2)-1}{2}} \quad (5.2)$$

$$= \sum_{h=1; h+=2}^{n-3} h \times 2^{\frac{n-h-3}{2}} \quad (5.3)$$

$$= \sum_{h=1; h+=2}^{n-3} h \times 2^{\frac{n-h-3}{2}} \quad \square \quad (5.4)$$

$$= \sum_{h=1}^{\frac{n-2}{2}} (2h-1) \times 2^{\frac{n-(2h-1)-3}{2}} \quad (5.5)$$

$$= \sum_{h=1}^{\frac{n-2}{2}} (2h-1) \times 2^{\frac{n-2h-2}{2}} \quad (5.6)$$

$$= 2^{\frac{n}{2}} \sum_{h=1}^{\frac{n-2}{2}} (2h-1) \times 2^{-h-1} \quad (5.7)$$

$$= 2^{\frac{n}{2}} \sum_{h=1}^{\frac{n-2}{2}} \frac{2h-1}{2^{h+1}} \quad \blacksquare \quad (5.8)$$

$$= 3 \times 2^{\frac{n-2}{2}} - n - 1 \quad (5.9)$$

(b) Si $2 \nmid n$:

$$R(n) = \sum_{h=0; h+=2}^{n-3} P_A(h \times 2) \times P_B(3 \times n - h - 2) \quad (5.10)$$

$$= 2^{\frac{n-3}{2}} + \sum_{h=2; h+=2}^{n-3} h \times 2^{\frac{(n-h-2)-1}{2}} \quad (5.11)$$

$$= 2^{\frac{n-3}{2}} + \sum_{h=2; h+=2}^{n-3} h \times 2^{\frac{n-h-3}{2}} \quad \square \quad (5.12)$$

$$= 2^{\frac{n-3}{2}} + \sum_{h=1}^{\frac{n-3}{2}} 2h \times 2^{\frac{n-2h-3}{2}} \quad (5.13)$$

$$= 2^{\frac{n-3}{2}} + \sum_{h=1}^{\frac{n-3}{2}} h \times 2^{\frac{n-2h-1}{2}} \quad (5.14)$$

$$= 2^{\frac{n-3}{2}} + 2^{\frac{n-1}{2}} \sum_{h=1}^{\frac{n-3}{2}} \frac{h}{2^h} \quad \blacksquare \quad (5.15)$$

$$= 5 \times 2^{\frac{n-3}{2}} - n - 1 \quad (5.16)$$

```

1 def r(n):
2     if n % 2: # 2 // n
3         return - n - 1 + 5 * 2 ** ((n - 3) // 2)
4     else: # 2 | n
5         return - n - 1 + 3 * 2 ** ((n - 2) // 2)
6
7 def pa3(n):
8     np = [1]
9     for i in range(2, n + 1):
10         np.append(sum(np) + i + r(i))
11     return np[-1]

```

FIGURE 5.5 – Algorithme de calcul de $P_A(n \times 3)$ de complexité $\mathcal{O}(n^2)$ en 

5.4 Trajet $(n \times 3; (1, 2); (\lambda; \lambda))$

Proposition 5.2 (Trajet $(n \times 3; (1, 2); (\lambda; \lambda))$).

$$1. \quad \text{Trajet}(n \times 3; (1, 2); (\lambda; \lambda)) = \text{Trajet}(n \times 3; (n, 2); (\lambda; \lambda)) = 2P_B(3 \times n - 1)$$

$$2. \quad (a) \text{ Si } 2 \mid n : \quad \text{Trajet}(n \times 3; (1, 2); (\lambda; \lambda)) = 2^{\frac{n}{2}} \quad \square$$

$$(b) \text{ Si } 2 \nmid n : \quad \text{Trajet}(n \times 3; (1, 2); (\lambda; \lambda)) = 0 \quad \square$$

Démonstration. A partir de $(2; 1)$ on peut aller soit à droite soit à gauche. Si on part au milieu, les deux angles sont isolés et l'un des deux ne sera pas atteint. On considère seulement le cas où on part à droite puis on multiplie par 2. Dans ce cas le point de gauche sera forcément le point d'arrivée donc on aura $G'(3 \times n - 1)$ à parcourir de bord à bord. D'où :

$$\text{Trajet}(n \times 3; (1, 2); (\lambda; \lambda)) = 2P_B(3 \times n - 1)$$

■

5.5 Trajets depuis les milieux : Trajet $(n \times 3; (\lambda, 2); (\lambda; \lambda))$

Introduction

On définit les milieux comme les points de coordonnées $(2, \lambda)$. On définit :

$$2\text{Parcours}_M(n) = \text{Trajet}(n \times 3; (\lambda, 2); (\lambda; \lambda))$$

On divise par deux car cela correspond à la symétrie d'axe (Oy) , le changement de variable correspond à partir vers le haut ou le bas :

$$2(P_M(n, h) + P_M(n, n - h + 1)) = \text{Trajet}(n \times 3; (h, 2); (\lambda; \lambda))$$

$$\text{Parcours}_M(n) = 2 \sum_{h=1}^n P_M(n, h) + P_M(n, n - h + 1)$$

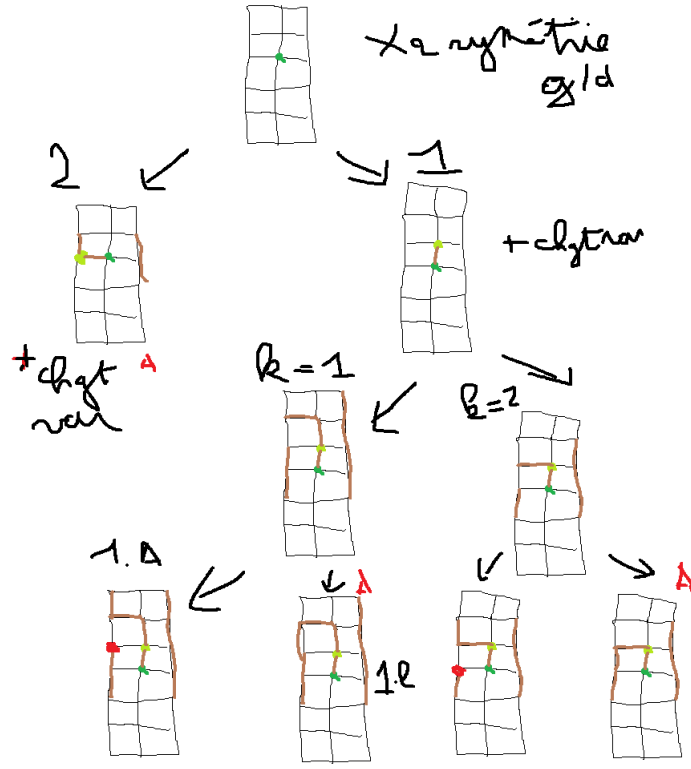


FIGURE 5.6 – Parcours depuis un milieu

Traduction en formules

En partant des dessins (cf Fig 5.6) qui sont même valide aux extrémités :

$$P_M(n, h) = P_1(n, h) + P_2(n, h)$$

$$P_2(n, h) = P_B(3 \times h - 1) P_A(3 \times n - h)$$

$$P_1(n, h) = P_{1a}(n, h) + P_{1l}(n, h) \quad (5.17)$$

$$= \sum_{k=1}^{h-2} P_B(3 \times n - h) P_B(3 \times k) + \sum_{k=0}^{h-2} P_B(3 \times n - h) P_A(3 \times k) \quad (5.18)$$

$$= \sum_{k=0}^{h-2} P_B(3 \times n - h) P_B(3 \times k) + \sum_{k=0}^{h-2} P_B(3 \times n - h) P_A(3 \times k) \quad \text{car } P_B(3 \times 0) = 0 \quad (5.19)$$

$$= P_B(3 \times n - h) \sum_{k=0}^{h-2} P_B(3 \times k) P_A(3 \times k) \quad (5.20)$$

$$(5.21)$$

$$P_M(n, h) = P_B(3 \times h - 1) P_A(3 \times n - h) + P_B(3 \times n - h) \sum_{k=0}^{h-2} P_B(3 \times k) P_A(3 \times k) \quad \blacksquare$$

Compatibilité avec Trajet $(n \times 3; (1, 2); (\lambda; \lambda))$ (cf 5.4)

$$P_M(n, 1) = P_B(3 \times 0) P_A(3 \times n - 1) + P_B(3 \times n - h) \sum_{k=0}^{-1} P_B(3 \times k) P_A(3 \times k) \quad (5.22)$$

$$= 0 \times 1 + P_B(3 \times n) \times 0 \quad (5.23)$$

$$= 0 \quad (5.24)$$

$$P_M(n, n) = P_B(3 \times n - 1) P_A(3 \times n - n) + P_B(3 \times n - n) \sum_{k=0}^{h-2} P_B(3 \times k) P_A(3 \times k) \quad (5.25)$$

$$= P_B(3 \times n - 1) \times 1 + 0 \times \sum_{k=0}^{h-2} P_B(3 \times k) P_A(3 \times k) \quad (5.26)$$

$$= P_B(3 \times n - 1) \quad (5.27)$$

Or

$$\text{Trajet}(n \times 3; (1, 2); (\lambda; \lambda)) = \text{Trajet}(n \times 3; (n, 2); (\lambda; \lambda)) = 2(P_M(n, n) + P_M(n, 1)) \quad (5.28)$$

$$= 2P_B(3 \times n - 1) \quad \blacksquare \quad (5.29)$$

Optimisation : symétries et théorème parité

- Si $2 \mid n$:

$$\text{Parcours}_M(n) = \sum_{h=1; 2 \nmid h}^n P_M(n, h) + P_M(n, n - h + 1) + P_M(n, h + 1) + P_M(n, n - (h + 1) + 1) \quad (5.30)$$

$$= \sum_{h=1; 2 \nmid h}^{n-1} P_M(n, h) + P_M(n, n - h + 1) + P_M(n, h + 1) + P_M(n, n - h) \quad (5.31)$$

$$= \sum_{h=1; 2 \nmid h}^{n-1} P_1(n, h) + P_2(n, n - h + 1) + P_2(n, h + 1) + P_1(n, n - h) \quad \blacksquare \quad (5.32)$$

- Si $2 \mid h$:

$$P_M(n, h) = P_1(n, h) + P_2(n, h) = P_2(n, h)$$

$$P_M(n, n - h + 1) = P_1(n, n - h + 1) + P_2(n, n - h + 1) = P_1(n, n - h + 1)$$

- Si $2 \nmid h$:

$$P_M(n, h) = P_1(n, h) + P_2(n, h) = P_1(n, h)$$

$$P_M(n, n - h + 1) = P_1(n, n - h + 1) + P_2(n, n - h + 1) = P_2(n, n - h + 1)$$

En utilisant la symétrie dans la longueur :

- Si $n \equiv 0[4]$:

$$\text{Parcours}_M(n) = 2 \sum_{h=1; 2 \nmid h}^{\frac{n}{2}-1} P_M(n, h) + P_M(n, n - h + 1) + P_M(n, h + 1) + P_M(n, n - h) \quad (5.33)$$

$$= 2 \sum_{h=1; 2 \nmid h}^{\frac{n}{2}-1} P_1(n, h) + P_2(n, n - h + 1) + P_2(n, h + 1) + P_1(n, n - h) \quad \blacksquare \quad (5.34)$$

– Si $n \equiv 2[4]$:

$$\text{Parcours}_M(n) = 2\text{Trajet}(n \times 3; (h, 2); (\lambda; \lambda)) \quad (5.35)$$

$$+ 2 \sum_{h=1; 2|h}^{\frac{n}{2}-2} P_M(n, h) + P_M(n, n-h+1) + P_M(n, h+1) + P_M(n, n-h) \quad (5.36)$$

$$= 2P_1\left(n, \frac{n}{2}\right) + 2P_2\left(n, \frac{n}{2}\right) \quad (5.37)$$

$$+ 2 \sum_{h=1; 2|h}^{\frac{n}{2}-2} P_1(n, h) + P_2(n, n-h+1) + P_2(n, h+1) + P_1(n, n-h) \quad \blacksquare \quad (5.38)$$

• Si $2 \nmid n$, grâce au théorème parité :

$$\text{Parcours}_M(n) = \sum_{h=2; 2|h}^{n-1} P_M(n, h) + P_M(n, n-h+1) + P_M(n, h+1) + P_M(n, n-(h+1)+1) \quad (5.39)$$

$$= \sum_{h=2; 2|h}^{n-1} P_M(n, h) + P_M(n, n-h+1) \quad (5.40)$$

$$= \sum_{h=2; 2|h}^{n-1} P_1(n, h) + P_2(n, h) + P_1(n, n-h+1) + P_2(n, n-h+1) \quad (5.41)$$

– Si $2 \mid h$:

$$P_M(n, h) = P_1(n, h) + P_2(n, h)$$

$$P_M(n, n-h+1) = P_1(n, n-h+1) + P_2(n, n-h+1)$$

– Si $2 \nmid h$:

$$P_M(n, h) = P_1(n, h) + P_2(n, h) = 0$$

$$P_M(n, n-h+1) = P_1(n, n-h+1) + P_2(n, n-h+1) = 0$$

En utilisant la symétrie dans la longueur :

– Si $n \equiv 1[4]$:

$$\text{Parcours}_M(n) = 2 \sum_{h=2; 2|h}^{\frac{n-1}{2}} P_M(n, h) + P_M(n, n-h+1) \quad (5.42)$$

$$= 2 \sum_{h=2; 2|h}^{\frac{n-1}{2}} P_1(n, h) + P_2(n, h) + P_1(n, n-h+1) + P_2(n, n-h+1) \quad \blacksquare \quad (5.43)$$

– Si $n \equiv 3[4]$:

$$\text{Parcours}_M(n) = 2P_M\left(n, \frac{n+1}{2}\right) \quad (5.44)$$

$$+ 2 \sum_{h=2; 2|h}^{\frac{n-1}{2}} P_M(n, h) + P_M(n, n-h+1) \quad (5.45)$$

$$= 2P_1\left(n, \frac{n+1}{2}\right) + 2P_2\left(n, \frac{n+1}{2}\right) \quad (5.46)$$

$$+ 2 \sum_{h=2; 2|h}^{\frac{n-1}{2}} P_1(n, h) + P_2(n, h) + P_1(n, n-h+1) + P_2(n, n-h+1) \quad \blacksquare \quad (5.47)$$

5.6 Trajets depuis les côtés : Trajet $(n \times 3; (\lambda, 1); (\lambda; \lambda))$

Introduction

$$\text{Parcours}_C(n) = \text{Trajet}(n \times 3; (\lambda, 1); (\lambda; \lambda)) = \text{Trajet}(n \times 3; (\lambda, 3); (\lambda; \lambda))$$

$$P_{C\uparrow}(n, h) + P_{C\uparrow}(n, n-h+1) = \text{Trajet}(n \times 3; (h, 1); (\lambda; \lambda))$$

Traduction en formules

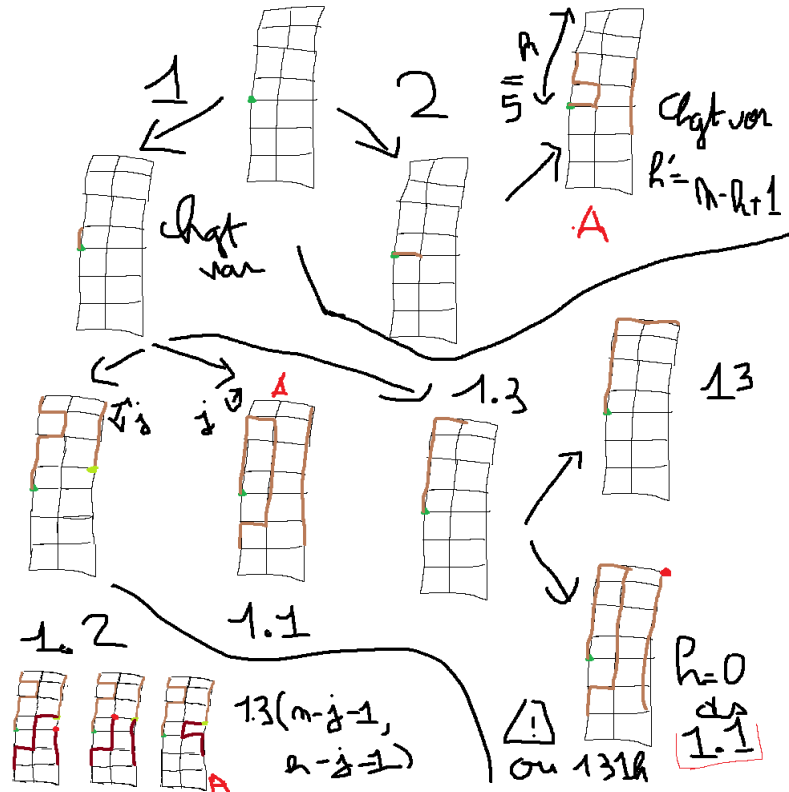


FIGURE 5.7 – Parcours depuis un côté

En partant des dessins qui sont même valides aux extrémités :

$$\text{Parcours}_C(n) = 2P_A(n \times 3) + \sum_{h=2}^{n-1} P_{C\uparrow}(n, h) + P_{C\uparrow}(n, n-h+1)$$

$$P_{C\uparrow}(n, h) = P_1(n, h) + P_2(n, h)$$

$$P_2(n, h) = P_B(3 \times h - 2) P_A(3 \times n - h)$$

$$P_1(n, h) = P_{1.1}(n, h) + P_{1.2}(n, h) + P_{1.3}(n, h)$$

$$P_{1.1}(n, h) = \sum_{k=0}^{h-2} P_B(3 \times n - h - 1) P_A(3 \times k) = P_B(3 \times n - h - 1) \sum_{k=0}^{h-2} P_A(3 \times k)$$

$$P_{1.2}(n, h) = \sum_{k=1}^{h-3} P_B(3 \times k) P_{1.3}(n - k - 1, h - k - 1)$$

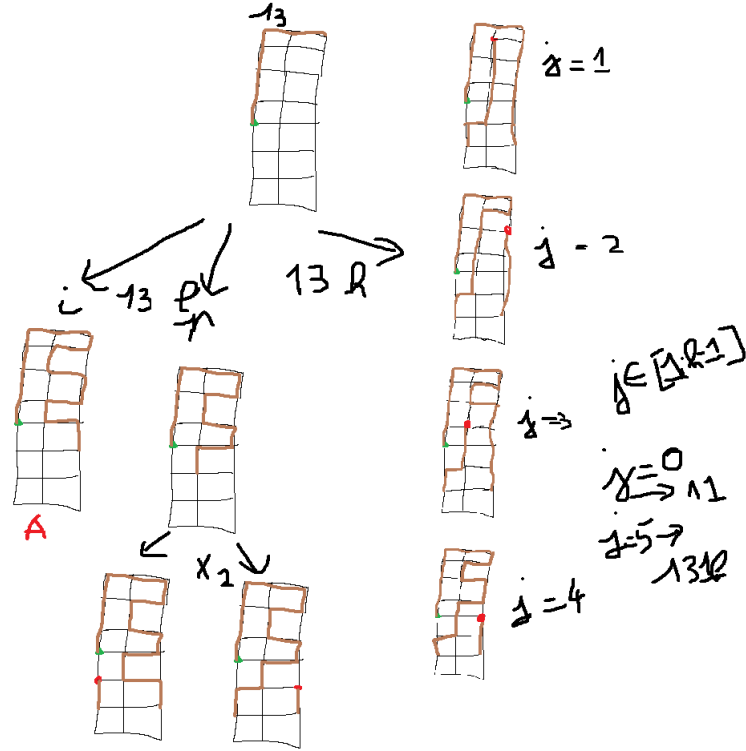


FIGURE 5.8 – Parcours 1.3.1

$$P_{1.3}(n, h) = P_{1.3.l}(n, h) + P_{1.3.h}(n, h) \quad (5.48)$$

$$= P_{1.3.l}(n, h) + \sum_{k=1}^{h-1} P_B(3 \times n - h - 1) \quad (5.49)$$

$$= P_{1.3.l}(n, h) + \sum_{k=1}^{h-1} P_B(3 \times n - h - 1) \quad (5.50)$$

$$= P_{1.3.l}(n, h) + (h - 1) P_B(3 \times n - h - 1) \quad (5.51)$$

$$(5.52)$$

$$P_{1.3.l} \left| \begin{array}{l} \text{Si } 2 \mid h; \\ \text{Si } 2 \nmid h; \end{array} \right. \begin{array}{l} P_{1.3.l}(n, h) = 2P_B(3 \times n - h - 1) \\ P_{1.3.l}(n, h) = P_A(3 \times n - h) \end{array}$$

Optimisation

- Si $2 \mid n$:

- Si $2 \mid h$:

$$P_{1.3}(n, h) = 2P_B(3 \times n - h - 1) + (h - 1)P_B(3 \times n - h - 1) = (h + 1)P_B(3 \times n - h - 1)$$

- Si $2 \nmid h$:

$$P_{1.3}(n, h) = P_A(3 \times n - h)$$

- Si $2 \nmid n$:

- Si $2 \mid h$:

$$P_{1.3}(n, h) = 0$$

- Si $2 \nmid h$:

$$P_{1.3}(n, h) = P_A(3 \times n - h) + (h - 1)P_B(3 \times n - h - 1)$$

Chapitre 6

Algorithme en

```
1 import numpy as np
2 from numba import jit
3
4
5 @jit(nopython=True)
6 def count_in_matrice(matrice, x, y, n):
7     if not n:
8         return 1
9     else:
10         result = 0
11         for dx, dy in (1, 0), (-1, 0), (0, 1), (0, -1):
12             if matrice.shape[0] > x + dx >= 0 and matrice.shape[1] > y + dy
13                 >= 0 and (not matrice[x + dx][y + dy]):
14                 # print(x + dx, y + dy)
15                 matrice[x + dx][y + dy] = True
16                 result += count_in_matrice(matrice, x + dx, y + dy, n - 1)
17                 matrice[x + dx][y + dy] = False
18         return result
19
20 def walkeur(matrice, x, y):
21     matrice[x][y] = True
22     r = count_in_matrice(matrice, x, y, matrice.shape[0] * matrice.shape[1]
23         - 1)
24     matrice[x][y] = False
25     return r
```

FIGURE 6.1 – Algorithme de dénombrement des parcours depuis un point en 

```

1 def count_path(Lx, ly):
2     """Fait tt les points de depart possibles"""
3     nodes = reheur(Lx, ly)
4     total = 0
5     if Lx % 2: # impair
6         if ly % 2: # impair
7             for x in range(0, Lx // 2): # angle * 4
8                 for y in range(0, ly // 2):
9                     if not (x + y) % 2: # th parite
10                        total += 4 * walkeur(coord_value(x, y, ly), Lx, ly,
11                                                nodes)
12                for y in range(0, ly // 2): # reste en x
13                    if not (Lx // 2 + y) % 2:
14                        total += 2 * walkeur(coord_value(Lx // 2, y, ly), Lx,
15                                                ly, nodes)
16                for x in range(0, Lx // 2): # reste en y
17                    if not (ly // 2 + x) % 2:
18                        total += 2 * walkeur(coord_value(x, ly // 2, ly), Lx,
19                                                ly, nodes)
20                if not (Lx // 2 + ly // 2) % 2: # milieu
21                    total += walkeur(coord_value(Lx // 2, ly // 2, ly), Lx, ly,
22                                        nodes)
23        else: # ly pair
24            for x in range(0, Lx // 2): # angle * 4
25                for y in range(0, ly // 2):
26                    total += 4 * walkeur(coord_value(x, y, ly), Lx, ly,
27                                            nodes)
28            for y in range(0, ly // 2): # reste en x
29                total += 2 * walkeur(coord_value(Lx // 2, y, ly), Lx, ly,
30                                        nodes)
31        else: # Lx pair
32            if ly % 2: # impair
33                for x in range(0, Lx // 2): # angle * 4
34                    for y in range(0, ly // 2):
35                        total += 4 * walkeur(coord_value(x, y, ly), Lx, ly,
36                                                nodes)
37                for x in range(0, Lx // 2): # reste en y
38                    total += 2 * walkeur(coord_value(x, ly // 2, ly), Lx, ly,
39                                            nodes)
39            else:
40                for x in range(0, Lx // 2): # angle * 4
41                    for y in range(0, ly // 2):
42                        total += 4 * walkeur(coord_value(x, y, ly), Lx, ly,
43                                                nodes)
44    return total

```

FIGURE 6.2 – Algorithme de dénombrement des parcours pour un graphe en 

c	Nb parcours	Temps (s)
1	1	2,980E-05
2	8	3,10E-05
3	40	4,68E-05
4	552	0,448
5	8 648	0,0109
6	458 696	3.656
7	27 070 560	1808.915
8	6 046 626 568	?
9	1 490 832 682 992	?
10	1 460 089 659 025 264	?
11	1 573 342 970 540 617 696	?
12	6 905 329 711 608 694 708 440	?
13	33 304 011 435 341 069 362 631 160	?
14	663 618 176 813 467 308 855 850 585 056	?
15	14 527 222 735 920 532 980 525 200 234 503 048	?
16	1 325 269 434 769 959 586 477 628 202 755 977 573 768	?
17	132 857 989 478 318 939 938 880 239 159 473 615 225 331 080	?

FIGURE 6.3 – Résultats de l'algorithme et de l'OEIS

Table des figures

1.1	Représentation du graphe 4×5	3
1.2	Représentation de Ω pour $G(2 \times 2)$	3
1.3	Représentation de Ω pour $G(3 \times 3)$	4
2.1	Application du corollaire 2.1.1 au graphe carré 5×5	6
2.2	Réduction au parcours du sous-graphe $G'(L \times l - 1)$	7
2.3	Réduction au parcours du sous-graphe $G'(L \times l - 3)$	8
2.4	Réduction au parcours du sous-graphe $G'(L \times l - 3)$ avec arrivée contrainte	8
4.1	Parcours direct en U	10
4.2	Réduction au parcours du sous-graphe $G'(n - 1 \times 2)$	10
4.3	Réduction au parcours de deux nouveaux sous-graphes : $G'(n - h \times 2)$ et de $G'(h - 1 \times 2)$	11
5.1	Deux parcours pour $P_B(3 \times n)$	12
5.2	Reduction à $P_O(n - 3 \times 3)$	13
5.3	Reduction à $P_B(n - 3 \times 3)$	13
5.4	Parcours restants après réduction par théorème réducteur	16
5.5	Algorithme de calcul de $P_A(n \times 3)$ en Python	18
5.6	Parcours depuis un milieu	19
5.7	Parcours depuis un côté	22
5.8	Parcours 1.3.1	23
6.1	Algorithme de dénombrement des parcours depuis un point en Python	25
6.2	Algorithme de dénombrement des parcours pour un graphe en Python	26
6.3	Nombre de parcours	27

Table des matières

1	Introduction	3
1.1	Définitions	3
2	Cas d'un graphe rectangle quelconque	5
2.1	Parité	5
2.1.1	Lemme : répartition de la parité	5
2.1.2	Théorème parité	5
2.1.3	Application du théorème parité aux graphes impairs	6
2.2	Théorème réducteur	6
3	Graphe $n \times 1$	9
4	Graphe $n \times 2$	10
4.1	Nombre de parcours à partir d'un angle	10
4.2	Parcours depuis un côté	11
4.3	Conclusion	11
5	Graphe $n \times 3$	12
5.1	$P_B(3 \times n)$	12
5.2	$P_B(n \times 3); P_O(n \times 3)$	13
	Propriétés	13
	Démonstrations	13
5.3	$P_A(n \times 3)$	14
	Propriétés	14
	Démonstrations	15
	Algorithme en Python	18
5.4	Trajet $(n \times 3; (1, 2); (\lambda; \lambda))$	18
5.5	Trajets depuis les milieux : Trajet $(n \times 3; (\lambda, 2); (\lambda; \lambda))$	18
	Introduction	18
	Traduction en formules	19
	Compatibilité avec Trajet $(n \times 3; (1, 2); (\lambda; \lambda))$ (cf 5.4)	19
	Optimisation : symétries et théorème parité	20
5.6	Trajets depuis les côtés : Trajet $(n \times 3; (\lambda, 1); (\lambda; \lambda))$	22
	Introduction	22
	Traduction en formules	22
	Optimisation	24
6	Algorithme en Python	25