Problème 2 : Mélodie des Hirondelles

$M\forall s^2 ili\forall$

3 mai 2022

Résumé

Nous étudions d'abord le cas où tous les oiseaux sont des hirondelles, et nous traitons toutes les questions (1 à 3), en nous concentrant dans les questions 3)c) et 3)d) sur le cas où le thème de départ est le plus éloigné possible du thème le plus aigu.
Puis nous considérons le cas avec une mésange (question 4), que nous traitons partiellement (reprise de la question 1).
Nous avons conçu des programmes en Python pour tester de nombreux cas et répondre de façon algorithmique à certaines questions.
Enfin, nous proposons quelques pistes de recherche. Notre preuve pour la question 2 repose sur un invariant (que nous n'avons pas réussi à généraliser en temps et en heure en présence de mésanges ou d'autres oiseaux).
Nous avons reconnu, dans la recherche du thème le plus aigu, un algorithme de tri, le tri à bulles, appliqué aux différences entre les positions : cela nous a aidé pour répondre à la question 3.
Concernant la 3)d) nous sommes initialement parti d'un programme Python pour pouvoir observer l'évolution du nombre de manières de lancer T graines. Nous avons finis par retrouver une formule générale pour cette séquence sur OEIS et avons explorer les objets combinatoires qui ont menés à cette formule.

Table des matières

1 Notations et Définitions

Ce problème met initialement en jeu des hirondelles posées sur des câbles électriques numérotés de bas en haut de $-k$ à k, avec $k \geqslant 0$. Lorsqu'Elaïa jette une graine sur une hirondelle i avec $i \in [[1, n]]$ alors celle-ci migrera vers un autre câble défini par $v_i = u_{i-1} + u_{i+1} - u_i$ où v_i représente donc la position finale de l'oiseau, u_{i-1} la position de l'oiseau de derrière et u_{i+1} celle de l'oiseau de devant.

Un thème noté $U_{1 \leqslant k \leqslant n}$ avec $k \in [[1; n]]$ est un n-uplet contenant la ou les valeurs des câbles se trouve les/la n hirondelle(s) indexées à l'aide de k. De ce fait, un thème existe à partir du moment où on a au minimum une hirondelle.
Un détail très important est que, lorsque n est fini, il existe un nombre limité de valeurs que i puisse prendre ; il existe donc un nombre limité d'hirondelles que l'on puisse indexer en indice de cette suite. Dans ce cas là on décide par convention que $u_0 = u_{n+1} = 0$.

Définition 1. *On note : $U \overset{i}{\longleftrightarrow} V$ la transformation qui consiste à donner une graine à l'hirondelle i et dans ce cas seule celle-ci se déplace à l'aide de la formule $v_i = u_{i-1} + u_{i+1} - u_i$*

Remarque : On ne peut lancer les graines que une par une, où V_i désigne la position après la graine et U_i,U_{i-1},U_{i+1} les positions de l'oiseau et de ses oiseaux adjacents avant que la graine soit lancée.

Définition 2. *On dit que des thèmes sont associés à un autre thème s'ils sont obtensibles à partir de celui-ci en lançant une ou plusieurs graines sur certains oiseaux.*

Définition 3. *Lorsque le nombre de câble k est fini, alors il se peut que lancer une graine provoque la migration d'un oiseau sur un câble qui a une valeur supérieure à k alors l'oiseau part du thème. On dira d'un thème dans lequel il est possible de faire partir les hirondelle qu'il n'est pas définitif. Dans le cas inverse, si peut importe le nombre de graines lancées aucun oiseau ne part du thème alors on dira qu'il est définitif.*

Définition 4. *Une permutation consécutive ou élémentaire est une permutation entre deux éléments consécutifs. Elle peut se faire avec un élément consécutif de n'importe quel côté donc on précisera à chaque fois les deux éléments concernés.*

Définition 5. *Lorsqu'on note $U_{i<n}$ ou plus généralement $i < n$ en indice c'est qu'on considère tout les éléments entre i et n.*

2 Question 1

2.1 *n hirondelles fini et k câbles infini*

Dans cette partie on cherche à établir l'ensemble des thèmes obtenables pour un nombre infini de câbles et d'hirondelles tel que, à chaque fois, au départ, une seule hirondelle se trouve sur le câble n°1 et les autres sur le câble n°0.

2.1.1 Avec n = 1 :

Les thèmes obtensibles avec une hirondelle sur le fil numéro 1 sont :

$$(1) \xleftrightarrow{1} (-1)$$

1 étant la position initiale et lorsqu'on lance une graine l'hirondelle effectue un changement de câble pour migrer vers le câble $v_1 = u_0 + u_2 - u_1$

Propriété 1 (Involution de transformation et symétrie centrale/rotation de 180°)**.** *On est dans le cas d'une involution ; si on relance une graine sur l'oiseau en position -1 alors il revient en position 1 . Dans ce cas là c'est tout simplement car $v_i = -u_i$ comme $u_{i+1} = u_{i-1} = 0$. Ce qui correspond aussi à une symétrie centrale ou bien une rotation de 180° autour du câble 0 au niveau de la position de l'hirondelle 1.*
Remarque : On aura donc dans tout nos graphes des "double flèches" : $U \xleftrightarrow{i} V$

Propriété 2 (Comportement des Hirondelles en bordure)**.** *Lorsqu'on lance une graine sur une hirondelle sur le câble n°0 et qu'elle est bordée par un 0 et un 1 ou -1 elle prendra une de ces deux valeurs.*

Démonstration. Ici c'est encore une fois parce que $u_{i+1} = 0$ ou $u_{i-1} = 0$ et $u_i = 0$ donc v_i prendra la seule valeur définie parmis u_{i+1} et u_{i-1} □

2.1.2 Avec n = 2 :

On obtient six thèmes possibles qui forment un cycle :

$$\begin{tikzcd}
(1,0) \arrow[r, leftrightarrow, "2"] \arrow[d, leftrightarrow, "1"'] & (1,1) \arrow[r, leftrightarrow, "1"] & (0,1) \arrow[d, leftrightarrow, "2"] \\
(-1,0) \arrow[r, leftrightarrow, "2"] & (-1,-1) \arrow[r, leftrightarrow, "1"] & (0,-1)
\end{tikzcd}$$

À partir des observations on déduit de nouvelles propriétées et définitions ;

Définition 6 (Symétrie centrale entre les thèmes constitués de valeurs strictement négatifs et positifs)**.** *Si on a $U \xleftrightarrow{i} V$ alors on a : $-U \xleftrightarrow{i} -V$*
Et même plus généralement pour tout $a \in \mathbb{Z}$, $aU \xleftrightarrow{i} aV$

Propriété 3 (Involution de transformation : cas plus complexe)**.** *Donner une graine deux fois de suite à la même hirondelle ramène au cas initial.*

Démonstration. En effet, dans ce cas on la suite de transformation suivantes : $U \xleftrightarrow{i} V \xleftrightarrow{i} W$
Pour les $j \neq i$ on a $u_j = v_j = w_j$.
$v_i = u_{i+1} + u_{i-1} - u_i$ et $w_i = v_{i+1} + v_{i-1} - v_i$
Donc $w_i = u_{i+1} + u_{i-1} - (u_{i+1} + u_{i-1} - u_i) = u_i$ donc $W = U$ □

2.1.3 Avec n = 3 :

Sur les thèmes musicaux positifs, on a le graphe suivant :

$$
\begin{tikzcd}
(1,0,0) \arrow[loop left, "3"] \arrow[r, leftrightarrow, "2"] & (1,1,0) \arrow[r, leftrightarrow, "3"] \arrow[d, leftrightarrow, "1"] & (1,1,1) \arrow[loop right, "2"] \arrow[d, leftrightarrow, "1"] & \\
 & (0,1,0) \arrow[r, leftrightarrow, "3"] & (0,1,1) \arrow[r, leftrightarrow, "2"] & (0,0,1) \arrow[loop right, "1"]
\end{tikzcd}
$$

En utilisant la propriété 3 on aura le même graphe sur les thème musicaux négatifs : $(-1,0,0),(-1,-1,0),(-1,-1,-1),(0,-1,-1),(0,0,-1),(0,-1,0)$ soit douze thèmes obtensibles au total.

Propriété 4 (Symétrie Axiale)**.** *En plus d'une symétrie centrale, il y aussi une symétrie axiale. C'est-à-dire que si une Hirondelle est bordé et sur les mêmes valeurs qu'une autre Hirondelle mais dans un sens opposée alors elles auront exactement le même comportement si on leur lance une graine.*

Démonstration. Si on jette une graine à une première hirondelle i bordée d'un côté par une valeur A et de l'autre par une valeur B et à une deuxième hirondelle j sur un câble de même valeur que l'hirondelle i bordée cette fois-ci de l'autre côté par A et B alors on obtiendra la même valeur. Ceci nous vient du fait que $u_{i+1} = u_{j-1}$, $u_{i-1} = u_{j+1}$ et $u_i = u_j$ on a finalement $v_i = u_{i+1} + u_{i-1} - u_i = u_{j-1} + u_{j+1} - u_i = v_j$ simplement par commutativité de l'addition. □

2.2 Cas Général avec un nombre infini d'hirondelles et de câbles

On retient quelques observations et conjectures des phases précèdentes :

1. Supposition n° 1 : Il n'y a aucuns thèmes qui partagent à la fois des (1) et (-1)
2. Supposition n° 2 : Il n'y a jamais, lorsque plusieurs oiseaux en 1 ou -1 sont présents dans un thème, des oiseaux en 0 qui les séparent.

Finalement en se basant sur ces observation, on émet l'hypothèse les thèmes obtenables sous ces conditions ne sont que des thèmes constitués de $0, 1, -1$ et que les 1 et -1 ne peuvent être présents simultanément et sont regroupés (ils ne peuvent pas être séparés par des 0).

Théorème 1. *Elaïa ne peut obtenir que des thèmes constitués de* $0, 1, -1$ *et que les* 1 *et* -1 *ne peuvent être présents simultanément et sont regroupés (ils ne peuvent pas être séparés par des* 0*).*

Démonstration par récurrence. Pour pouvoir démontrer ce théorème et répondre à la première question, nous effectuons une preuve par récurrence sur le nombre de graines pour vérifier que peut importe le nombre de graine que nous jetons les thèmes obtenus respectent le théorème 1. □

On note g le nombre de graines que l'on lance sur les hirondelles. Notre hypothèse est :
$P(g)$: les thèmes obtenables en lançant g graines sont constitués de $0, 1, -1$, les 1 et -1 sont regroupés sans être séparés par des 0, il ne peut pas y avoir de 1 et -1 en même temps.

Initialisation : Lorsque $g = 0$, il y a un oiseau sur le câble n°1 et tout les autres sur le câble n°0 selon la consigne. La propriété $P(0)$ est bien satisfaite.

Hérédité : Maintenant on suppose que $P(g)$ est vraie et on cherche à démontrer $P(g+1)$ dans ce cas là. On fait une distinction de cas selon la position de l'hirondelle à qui on lance la graine $g + 1$ ainsi que les positions de ses voisines : u_{i-1}, u_i, u_{i+1}. Le thème obtenu après g graines devrait donc ressembler par exemple à $0, 1, 0$ ou $0, -1, 0$ ou $0, 1, 1, 1, 0$ ou $0, -1, -1, -1, 0$.
Pour la graine $g + 1$ on peut la lancer sur une hirondelle en 0, ou une hirondelle 1 entouré de 0 ou une hirondelle en 1 entouré de 1 ce qui nous donne :

Si on lance sur le premier oiseau : $(0, 1, 0) \xleftrightarrow{1} (1, 1, 0)$ On respecte bien la propriété.
Si on lance sur le deuxième oiseau : $(0, 1, 0) \xleftrightarrow{2} (0, -1, 0)$ On respecte bien la propriété.
Maintenant si on lance une graine sur une hirondelle sur le câble 1 entouré d'hirondelles sur le câble 1 on a alors : $(0, 1, 1, 1, 0) \xleftrightarrow{3} (0, 1, 1, 1, 0)$ On respecte bien la propriété.
Et enfin si on lance la graine sur une hirondelle en bordure d'une autre hirondelle en 0 et d'une autre hirondelle en 1 on a : $(0, 1, 1, 0) \xleftrightarrow{2} (0, 0, 1, 0)$ On respecte bien la propriété.

Finalement, par symétrie axiale (Propriété 4) on trouve que cela fonctionne aussi lorsque l'hirondelle est bordée cette fois-ci par la droite par un 0 et par la gauche par un 1 (dans l'exemple on a seulement traité lorsqu'elle était bordée par la gauche par un 0 et par la droite par un 1). Et par symétrie centrale (Définition 1) on trouve aussi que tout les cas avec des -1 fonctionnent aussi car ils se comportent comme avec des 1.

On en déduit que $P(g + 1)$ est vrai, id est ajouter une graine à n'importe quel oiseau d'un thème respectant $P(g)$ ne le fait pas sortir des conditions qu'on a initialement fixées.
$P(0)$ étant aussi vrai, on conclut que $\forall g \in N$, $P(g)$ est vrai et que donc les thèmes obtenables sont constitués de $0, 1, -1$ et que les 1 et -1 lorsqu'ils sont présents sont regroupés et ne peuvent pas être séparés de 0. Il ne peut pas y avoir de 1 et -1 en même temps.

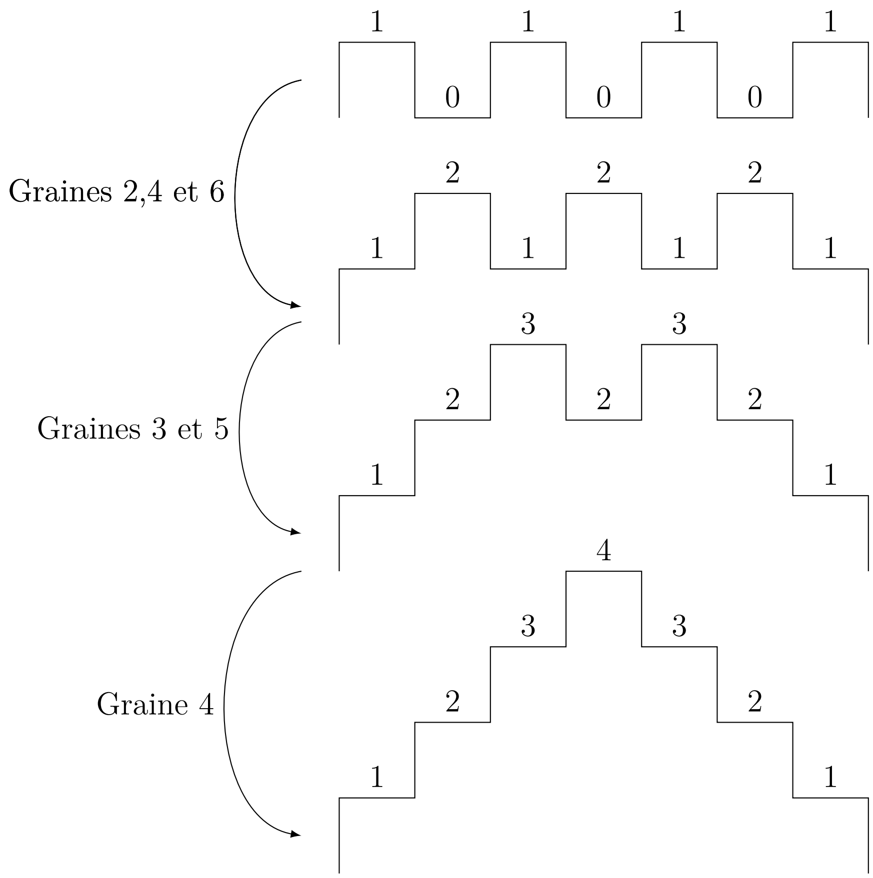
3 Question 2 : Thèmes définitifs

Cette question en implique une autre intermédiaire; jusqu'à quelle altitude les hirondelles pourront-elles monter ?
En effet pour k câbles finis si une Hirondelle migre vers un câble supérieur à k alors elle sort et part définitivement. Par symétrie centrale des valeurs des câbles et du comportement des hirondelles ce raisonnement fonctionne aussi pour des hirondelles qui migrerait sur une valeur de câble inférieur à $-k$.

A partir de cette question on va essayer de trouver un moyen de prédire pour chaque thème le maximum ou un majorant
On commence à observer un thème créé pour maximiser U_{i+1} et U_{i-1} car $V_i = U_{i-1} + U_{i+1} - U_i$ lorsqu'on lance une graine sur i qui est celui-ci :

$$(1,0,1,0,1) \xleftrightarrow{1,2} (1,2,1,2,1) \xleftrightarrow{3} (1,2,3,2,1)$$

On conjecture d'abord que lorsque n est impair on peut monter jusqu'à $\frac{n+1}{2}$ et lorsqu'il est pair il monte jusqu'à $\frac{n}{2}$ pour ce genre de thème constitué initialement de deux valeurs. C'est malheureusement trop restrictif pour conclure sur l'ensemble des thèmes.
On modélise aussi la situation géométriquement avec des escaliers :



Lemme 1. *Comportement des "creux"*
Si on lance une graine à une Hirondelle entourée de deux valeurs plus hautes qu'elle, alors sa nouvelle position sera plus haute que ses deux valeurs. On a obtenu une condition pour qu'on ne puisse plus faire monter : $\forall i, u_i \geqslant (u_{i-1} + u_{i+1})/2$.

Démonstration. On a toujours $V_i = U_{i+1} + U_{i-1} - U_i$ donc si $u_i \geqslant (u_{i-1} + u_{i+1})/2$ alors V_i ne peut simplement pas dépasser U_i □

On estime donc que l'altitude maximale est atteinte lorsque le thème forme un "escalier optimal" c'est-à-dire qu'il monte le plus haut des deux côtés. Et que l'hirondelle au sommet de cet "escalier optimal" représente la maximum atteignable pour ce thème et tout ses thèmes associés.

Une manière plus algébrique de voir ce phénomène est d'observer l'écart entre chaque marche qui correspond à la différence entre deux oiseaux consécutifs.
Selon le Lemme 1, un oiseau prend beaucoup d'altitude lorsqu'on jette une graine dans un creux car le creux implique que $U_i \geqslant U_{i+1}$ et U_{i-1}. Et lorsqu'on fait ça on étend et agrandit notre escalier. La valeur du maximum obtensible d'un thème serait donc la valeur maximum du thème associé formant un escalier parfait.

A partir de toutes ces observations et toujours en quête d'un majorant ou un maximum qui fonctionne pour tout les thèmes - car se basant sur le thème hypothétique associé qui monte le plus - on va commencer à étudier les différences entre les membres consécutifs des thèmes. On recherche des tendances et des propriétés intéressantes dans celle-ci comme un majorant qui soit invariant pour un thème et tout ses thèmes associées dont le thème qui contient la valeur maximum par exemple qui est celui qui nous intéresse. Ce que nous allons finir par trouver après beaucoup d'observations. Nous allons même démontrer que c'est un maximum et conclure sur la question 2.

3.1 Changement de Perspective sur les thèmes

3.1.1 Définition Récursive de $U_{1\leqslant k\leqslant n}$

Lemme 2. *Toute Hirondelle $U_{1\leqslant k\leqslant n}$ a une définition récursive et est égale à* $U_k = (U_1 - U_0) + (U_2 - U_1) + ... + (U_k - U_{k-1}) = (U_n - U_{n+1}) + (U_{n-1} - U_n) + ... + (U_k - U_{k+1})$

Démonstration. Comme $U_0 = U_{n+1} = 0$ alors $U_1 = U_1 - U_0$ et $U_n = U_n - U_{n+1}$
Par la suite, on trouve que
$(U_1 - U_0) + (U_2 - U_1) = U_1 + U_2 - U_1 = U_2$ et
$(U_n - U_{n+1}) + (U_{n-1} - U_n) = U_n + U_{n-1} - U_n = U_{n-1}$
On peut continuer ainsi de suite pour toutes les valeurs de U. On conclut donc par récursivité le Lemme 2. □

3.1.2 Introduction du n+1 uplets de différences D

À partir de là on change de perspective sur les thèmes et on introduit pour chaque thème de n hirondelles un $n+1$ uplets D des différences associées à chaque membre du thème tel que :
$D_{0\leqslant i\leqslant n} = U_{i+1} - U_i$
Pour le thème $a = (2, 4, 5, 3, 3)$ on a $Da = (2, 2, 1, -2, 0, -3)$
En analysant plusieurs thèmes à travers leur D associé et en jetant des graines, on trouve plusieurs observations intéressantes : $a = (2, 4, 5, 3, 3)$
Remarque : on indexe D à partir de 0 tandis que U est toujours indexé à partir de 1

$$(2,4,5,3,3) \xleftrightarrow{3} (2,4,2,3,3)$$

On appelle $a' = (2, 4, 2, 3, 3)$ on trouve que $Da' = (2, 2, -2, 1, 0, -3)$ on remarque que $Da_i = Da'_i$ sauf pour $i = 2et3$ qui ont été permutées.

A partir de là on conjecture que que les valeurs des différences entre un D quelconque et un D' obtensible à partir de D en lançant des graines sont conservées. C'est-à-dire qu'elles restent toutes présentes mais simplement que leur ordre change.

On émet aussi l'hypothèse que lancer une graine sur une Hirondelle i revient à permuter les valeurs D_i et D_{i+1} sur le D' obtenu après cette graine.

3.2 Conservation des différences positives et négatives et permutations

Lemme 3. *Toutes les valeurs des différences sont conservées unes à unes entre un D quelconque et un D' obtensible à partir de D en lançant des graines qui provoque une permutation élémentaire.*

Démonstration. Les deux seules valeurs de D et D' qui seront impactées par cette graine sont celles qui contiennent la position U_i de l'hirondelle i qui reçoit la graine dans leur définition. On a $D_i = U_{i+1} - U_i$ et $D_{i-1} = U_i - U_{i-1}$.

On jette une graine sur l'hirondelle i pour obtenir dans la définition de D'_i et D'_{i-1} V_i à la place de U_i (avec on le rappelle $V_i = U_{i+1} + U_{i-1} - U_i$)
On a donc :

$$D'_i = U_{i+1} - V_i = U_{i+1} - U_{i+1} - U_{i-1} + U_i = U_i - U_{i-1} = D_{i-1}$$

et

$$D'_{i-1} = V_i - U_{i-1} = U_{i+1} + U_{i-1} - U_i - U_{i-1} = U_{i+1} - U_i = D_i$$

On conclut que

$$D'_i = D_{i-1}$$

et

$$D'_{i-1} = D_i$$

Lancer une graine provoque une permutation élémentaire. □

Corollaire 1. *Conservation des différences entre un thème et ses thèmes associés*
Si lancer une graine provoque une permutation entre le n+1 uplets D et D' et qu'on ne peut lancer les graines une par une alors tout n+1 uplets $D^{k'}$ obtenu après un nombre k graines contient toujours les mêmes éléments que le n+1 uplet D initial et tous les $D^{1 \leqslant i \leqslant k'}$ intermédiaires. Ils ne seront simplement pas forcément dans le même ordre.
On conclut qu'il y a conservation des différences entre un thème et ses thèmes associés.

3.3 L'invariant est Majorant !

À partir du Corollaire sur la conservation des différences, on peut établir que la formule $\sum_{k=0}^{n} |U_{i+1} - U_i|$ est un invariant pour tout thème et ses thèmes associés, car peu importe leur ordre, si on additionne la valeur absolue de tous les éléments on obtiendra la même chose.

Lemme 4 (Majorant)**.** $\frac{1}{2}\sum_{k=0}^{n} |U_{i+1} - U_i|$ *est un invariant qui majore la valeur maximum qu'un oiseau peut atteindre pour n'importe quel thème et tout ses thèmes associés.*

Démonstration. En utilisant le Lemme 2 sur la définition récursive des mésanges et avec l'inégalité triangulaire on trouve :

$$|U_{1\leqslant k\leqslant n}| \leqslant |U_1 - U_0| + .. + |U_k - U_{k-1}|$$

et

$$|U_{1\leqslant k\leqslant n}| \leqslant |U_{n+1} - U_n| + .. + |U_{k+1} - U_k|$$
$$\Longleftrightarrow 2|U_{1\leqslant k\leqslant n}| \leqslant \sum_{i=0}^{n} |U_{i+1} - U_i|$$
$$\Longleftrightarrow |U_{1\leqslant k\leqslant n}| \leqslant \frac{1}{2}\sum_{i=0}^{n} |U_{i+1} - U_i|$$

Par la suite on sait que $\frac{1}{2}$ est une constante et que la valeur $\sum_{k=0}^{n} |U_{i+1} - U_i|$ est invariante pour chaque thème et tous ses thèmes associés donc notre majorant est bien aussi invariant. □

3.4 Maximum et conclusion

Lemme 5. *Conséquence de la symétrie axiale sur D*
Par symétrie axiale on a que la somme des différences positives est égale à l'opposé de la somme des valeurs négatives. D prenant en compte les différences de $U_0 = 0$ à $U_{n+1} = 0$ on a donc la somme des différences positifs qui est égale à $V \in \mathbb{N}$ car c'est une somme d'entiers naturels et celle des différences négatives est égale à $-V \in \mathbb{Z}$ car c'est une somme d'entiers relatifs.
On en déduit que :

$$\sum_{i=0}^{n} |U_{i+1} - U_i| = 2V$$
$$<=> \frac{1}{2}\sum_{i=0}^{n} |U_{i+1} - U_i| = V$$

Démonstration. Pour que le majorant soit atteint il faut donc que $U_k = V$.
En utilisant la définition récursive de U_k on trouve que $(U_1 - U_0) + ... + (U_k - U_{k-1})$ doit être égale à V et donc que $D_{0\leqslant k-1} \leqslant 0$ or lancer une graine permet d'effectuer des permutations. Il est donc possible de lancer les graines de façon à trier toute les permutations positives de D_0 à D_{k-1} avec k suffisamment grand.
On conclut qu'il y a bien pour chaque thème, un thème associé où les différences positives sont toutes consécutives et placées au début tel que :

$$U_k = V = \frac{1}{2}\sum_{i=0}^{n} |U_{i+1} - U_i|$$

.
La valeur de notre majorant invariant étant toujours atteignable pour n'importe quel thème, on déduit que c'est un maximum. □

Théorème 2. *Thèmes Définitifs - Conclusion On appelle la valeur $M = \frac{1}{2}\sum_{i=0}^{n} |U_{i+1} - U_i|$ la valeur maximum atteignable par les hirondelles dans tous les thèmes associés à un thème général et selon le Lemme 2 sur la définition récursive des hirondelles, le Lemme 3 sur l'invariance des différences, le Lemme 4 sur le majorant et le Lemme 5 sur le maximum, on conclut que tout les thèmes avec k câbles tel que $k \leqslant M$ sont définitifs c'est à dire qu'Elaïa est certaine qu'aucune hirondelle ne partira. Les thèmes avec $k < M$ câbles ne seront quant à eux pas définitives et les hirondelles pourront partir.*

4 Question 3 : Thèmes Aigus

Propriété 5. *On dira qu'un thème musical* $(u'_i)_{1 \leqslant i \leqslant n}$ *est plus aigu qu'un autre* $(u_i)_{1 \leqslant i \leqslant n}$ *si pour chaque* $i \in [[1, n]]$*, on a* $u'_i \geqslant u_i$*.*

Dans l'exemple de la question précédente, on trouve comme thème le plus aigu évident par rapport à $1, 0, 1, 0, 1$ qui est $1, 2, 3, 2, 1$
Mais aussi, en partant de $2, 1, 5, 2, 1$:
On lance des graines de manière à atteindre le thème le plus aigu :
$2, 1, 5, 2, 1 \xleftrightarrow{2} 2, 6, 5, 2, 1 \xleftrightarrow{4} 2, 6, 5, 4, 1 \xleftrightarrow{1} 4, 6, 5, 4, 1 \xleftrightarrow{5} 4, 6, 5, 4, 3$

4.1 Question 3)a) : Tri à bulles

On remarque alors que lorsqu'un thème est aigu, les différences sont triées dans l'ordre décroissant et qu'à chaque graine lancée qui nous rapproche du thème le plus aigu une permutation "utile" est effectuée.

Remarque : on ne considère ici que des permutations qui échangent deux éléments consécutifs.
On reconnaît le tri à bulles.

Démonstration. On revient à la définition récursive de $U_{1 \leqslant k \leqslant n}$. On note U un thème lambda, D son n+1 uplets de différence, et on note U' le thème le plus aigu et D'.
Pour que $\forall k \in [[1, n]], U'_k \geqslant U_k$ il faut donc que :

$$(U'_1 - U'0) + ... + (U'_k - U'_{k-1}) \geqslant (U_1 - U_0) + ... + (U_k - U_{k-1})$$

$$<=> D'_k \geqslant D_k$$

Par la suite selon le Lemme 3 sur les conséquences des graines sur les différences on sait que lancer une graine provoque une permutation. Dans ce cas là on se retrouve dans une situation de tri à bulles. Il suffit juste d'envoyer les graines pour permuter les valeurs lorsque le couple consécutif que l'on considère est trié dans l'ordre croissant. En répétant cet algorithme et en itérant plusieurs fois sur l'ensemble des valeurs jusqu'à ce que $\forall i \in [[0; n]] D_i \geqslant D_{i<n}$
Nous sommes donc garantit que pour tout thème, si ce n'est pas déjà le plus aigu alors non seulement il existe un thème plus aigu mais aussi qu'il est obtenable en lançant des graines.

Remarque : Comme précisé dans le Lemme 5 sur la Conséquence de la symétrie axiale sur D, en considérant la somme des éléments positifs de D égale à V alors la somme des éléments négatifs est égale à $-V$. On rappelle aussi que comme un thème contient au minimum une hirondelle alors D contient au minimum $n + 1$ soit deux valeurs au minimum. □

4.2 Question 3)b) : Retrouver le thème le plus aigu

b) Si oui, quel est le thème musical le plus aigu, en fonction du thème de départ ?

Le thème le plus aigu en fonction d'un thème de départ est celui qui possède les différences entre u_i et u_{i+1} de ce thème de départ dans l'ordre décroissant.
Jeter une graine revient à permuter ces différences.
On réutilise

Propriété 6. *Afin de retrouver le thème associé à un n+1 uplets de différences D on peut simplement faire la réciproque de la différence c'est à dire la somme sachant que les bordures sont = 0 par convention.*

donc le tri à bulles pour pouvoir retrouver le thème le plus aigu.
On peut aussi trier manuellement les différences de D associé au thème avec notre algorithme de tri favori et utiliser la réciproque de la différence.

Pour $U = 1, 0, 1, 0, 1, 0, 1$ on a $D_{initial} = 1, -1, 1, -1, 1, -1, 1, -1$
On trouve à la main $D_{décroissant} = 1, 1, 1, 1, -1, -1, -1, -1$ on a donc $U_{aigu} = 1, 2, 3, 4, 3, 2, 1$

4.3 Question 3)c) : Minimum T

c) Si $(u_i)_{1 \leqslant i \leqslant n}$ désigne la position initiale des hirondelles, combien de graines $T((u_i)_{1 \leqslant i \leqslant n})$ Elaïa doit-elle alors utiliser au minimum pour obtenir ce thème musical le plus aigu ?

Conjecture n°1 :
$1, 2, 3, 4, 5 => 5, 4, 3, 2, 1$: le 1 bouge de 4 positions, le 2 de 2, le 3 de 0, le 4 de 2, le 5 de 4. En tout il faut donc 4+2+0+2+4=12 mouvements. Chaque permutation fait 2 mouvements donc il faut (au moins) 6 permutations.
Ce raisonnement est faux ! !

On remarque que le nombre de permutations se comporte différemment : 1 => 1 : 0 permutation
$1, 2 => 2, 1$: 1 permutation
$1, 2, 3 => 3, 2, 1$: 3 permutations
$1, 2, 3, 4 => 4, 3, 2, 1$: 6 permutations
$1, 2, 3, 4, 5 => 5, 4, 3, 2, 1$: 10 permutations
$1, 2, 3, 4, 5, 6 => 6, 5, 4, 3, 2, 1$: 15 permutations

Théorème 3. *Le minimum de permutations nécessaire, soit le nombre de graines que Elaïa doit utiliser pour passer d'un thème quelconque à son thème le plus aigu vaut* $T = C(2, n+1) = n(n + 1)/2$

Démonstration. On veut trier $a(1)...a(n)$ dans l'ordre décroissant. On compte le nombre N de (i, j) tels que $i < j$ et $a(i) < a(j)$. Ce nombre vaut 0 si la suite est déjà bien triée, et vaut au maximum $n(n - 1)/2$, ce qui se produit quand elle est triée dans l'ordre croissant.
A chaque fois qu'on fait une permutation élémentaire (entre deux éléments consécutifs), N varie de 1 (en plus ou en moins) car l'ordre d'un seul couple est modifié. Dans le tri à bulles, N décroît toujours de 1 (car on ne permute deux éléments consécutifs que quand ils ne sont pas bien ordonnés). Donc le cas le pire est celui de la suite triée dans l'ordre croissant, pour laquelle N va décroitre de 1 en 1 de $n(n - 1)/2$ à 0, le temps est donc $T = n(n - 1)/2$.

Revenons aux hirondelles. En général, on peut écrire :

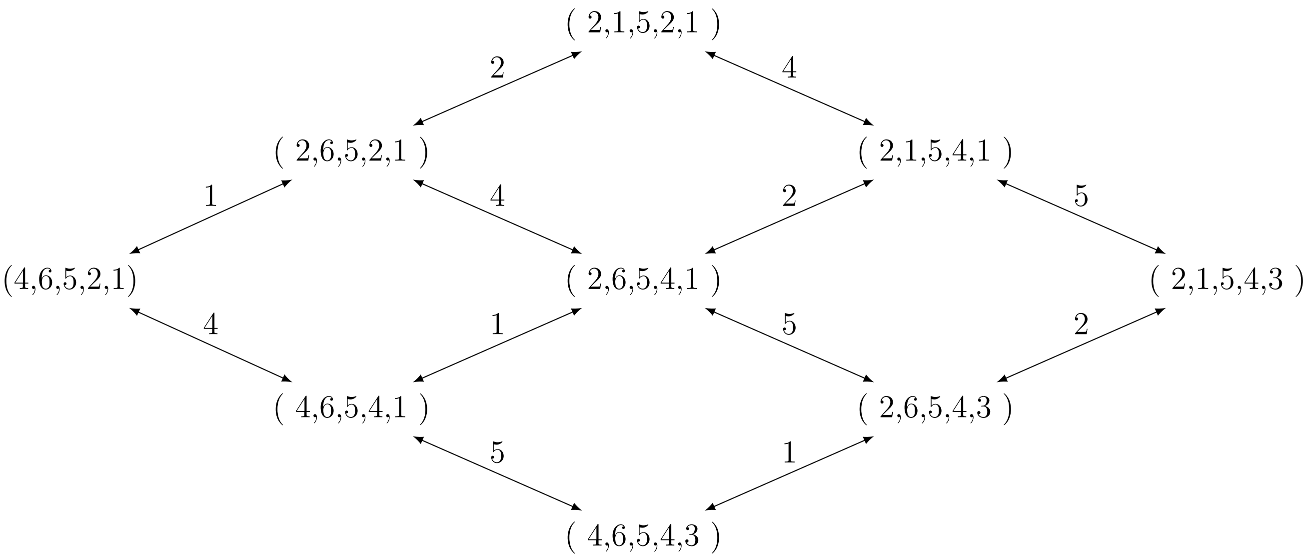
$$T((u_i)_{1 \leqslant i \leqslant n}) = \sum_{i=0}^{n} \mathrm{Card}\{j | i \leqslant j \leqslant n \text{ et } u_{i+1} - u_i < u_{j+1} - u_j\}.$$

Attention : si le nombre d'hirondelles est n, le nombre de différences est $n + 1$ (à cause des hirondelles fictives sur le fil 0 de part et d'autre). Donc le cas le pire est $T((u_i)_{1 \leqslant i \leqslant n}) = n(n+1)/2$.
□

4.4 Question 3)d) : Manière de lancer T graine

d) Dans les cas où Elaïa obtient le thème musical le plus aigu en lançant $T((u_i)_{1\leqslant i\leqslant n})$ graines, combien de suites de lancers différents lui permettent d'atteindre ce thème le plus aigu en $T((u_i)_{1\leqslant i\leqslant n})$ graines ?

26521 et 21541 sont deux thèmes qui peuvent être obtenus à partir de 21521. Aucun des deux n'est plus aigu que l'autre : on ne peut pas toujours comparer deux thèmes.



Dans cet exemple, il y a 6 chemins (= 2 parmi 4).
On s'intéresse désormais au n+1 uplet D et au manière d'effectuer les permutations dans différents cas pour qu'il soit trié dans l'ordre décroissant :
12 → 21 : 1 seule suite de permutations
$123 \to 321$: $123 \to 213 \to 231 \to 321 ou 123 \to 132 \to 312 \to 321$: 2 suites de permutations possibles.

$$\begin{tikzcd}
 & D=(1,2,3) \arrow[dl] \arrow[dr] & \\
D=(2,1,3) \arrow[d] & & D=(1,3,2) \arrow[d] \\
D=(2,3,1) \arrow[dr] & & D=(3,1,2) \arrow[dl] \\
 & D=(3,2,1) &
\end{tikzcd}$$

1234 → 4321 : 16 suites de permutations

Les suites de permutations fait à la main ne nous donnent pas assez de matière pour pouvoir correctement conjecturer. À partir de là je me mets à faire un programme Python qui nous permettra d'entrevoir plus de valeurs et peut-être de trouver une formule.

```python
from functools import lru_cache

def bubblesort(number):
    for i in range(len(number)):
        if int(number[i+1])<int(number[i]):
            number[i+1],number[i]=number[i],number[i+1]
    return number

def worst_bubble_situtation_gen(l):
    final = ""
    for i in range(int(l)):
        final += str(i+1)
    return final

def sort_number(number):
    liste = [i for i in number]
    liste.sort(reverse = True)
    return liste

@lru_cache(maxsize=None)
def count_permutations(number):
    if [i for i in number] == sort_number(number) :
        return 1
    else:
        final = 0
        for i in range(len(number)-1):
            if int(number[i+1]) > int(number[i]):
                final += count_permutations(number[0:i]+number[i+1]+
                                            number[i]+number[i+2::])
        return final

n=input("entrez un 'n' personnalisé sous forme de nombre")
print(
    "Pour n = 2 :", count_permutations("12"),"\n",
    "Pour n = 3 :", count_permutations("123"),"\n",
    "Pour n = 4 :", count_permutations("1234"),"\n",
    "Pour n = 5 :", count_permutations("12345"),"\n")

print("Pour n = 6 :", count_permutations("123456"),"\n")

print("Pour n =",n," : ", count_permutations(worst_bubble_situtation_gen(n)))
```

Le programme repose sur une boucle récursive et un cache pour l'optimisation qui va sauvegarder les données pour éviter des calculs superflus. On vérifie à chaque fois si le paramètre "number" rentré correspond à une suite triée (soit le thème musical le plus aigu dans le cadre du problème) et si ce n'est pas le cas alors on itère sur les permutations possibles et à chaque permutation en profondeur (voir le programme comme un graphe). À chaque niveau/étage de notre arbre, on itère de nouveaux sur toutes les permutations utiles et on incrémente la valeur "final" pour chaque nouvelle itération possibles qui correspondent donc à de suites de lançages de graines différentes. L'itération nous garantit notamment de ne pas effectuer 2 suites de lancers similaires.

La valeur finale correspond donc au nombre de manières d'effectuer T lancers pour un thème U de longueur n.

NB : Dans le programme, pour répondre à la question, on a délibérément choisi les pires cas pour les suites u_i (c'est-à-dire que les différences $u_i - u_{i-1}$ sont triés dans l'ordre croissant).

Grâce aux résultats de ce programme :

```
Pour n = 2 : 1
 Pour n = 3 : 2
 Pour n = 4 : 16
 Pour n = 5 : 768

Pour n = 6 : 292864

Pour n = 11  :  20568631998369717288960
```

Sur OEIS on a pu trouver une formule générale :

$$a(n) = \frac{C(n,2)!}{(1^{n-1} \times 3^{n-2} \dots (2n-3)^1}$$

$C(n,2)!$ est un majorant, il représente l'ensemble des permutations quelconques (donc pas forcément consécutives ce qui n'est pas possible dans le cas du tri à bulles/du lançage de graine) possibles. C'est un tirage sans remise des combinaisons de permutations possibles.
Exemple : $n = 3$, $C(n,2) = 3$ donc le majorant est $3! = 6$ alors que $a(3) = \dfrac{6}{(1^2 \times 3^1)} = 2$.
Les 6 possibilités sont :
(12)(13)(23)123 → 213 → 231 → 321 (tri à bulles)
(12)(23)(13)123 → 213 → (non consécutifs)
(13)(12)(23)123 → (non consécutifs)
(13)(23)(12)123 → (non consécutifs)
(23)(12)(13)123 → 132 → (non consécutifs)
(23)(13)(12)123 → 132 → 312 → 321(tri à bulles)

4.4.1 Pour approfondir la piste donnée par OEIS..

Après quelque recherche, on trouve que la formule correspond aussi au nombre de tableaux de Young d'une certaine forme.
On ne va pas beaucoup étayer cette partie car les résultats ne proviennent pas de notre équipe mais plutôt de la littérature combinatoire cela dit un aperçu est toujours intéressant dans la compréhension de la formule générale.

On note "SYT" abréviation de Standard Young Tableaux
Un SYT est un tableau où si on note $f_{1,\lambda}$ le nombre de colonne λ à la première ligne alors on a : $f_{i,\lambda} \geqslant f_{i<n,\lambda}$ où n désigne le nombre total de lignes.

On a aussi pour chaque colonne et chaque ligne que l'élément antécédent majore strictement tout les éléments qui le succède.
Si on note les cases du tableaux $c_{i,j}$ avec i pour l'indexation de la ligne et j de la colonne alors on a $c_{i,j} > c_{i<n+1,j}$ et $c_{i,j} > c_{i,j<\lambda_i+1}$. où λ_i désigne le nombre de colonnes dans ladite ligne.
Attention : c'est seulement la colonne ou la ligne qui et pas la diagonale ou tout les élements dans le rectangle formé par la colonne et la ligne qui sont pris en compte c'est à dire qu'on peut avoir $c_{i,j} > c_{i<n+1,j<\lambda_i+1}$ mais ce n'est pas forcément le cas.

Par la suite on s'intéresse à la "hook length formula" qui est une formule qui à partir de la forme d'un tableau standard de young arrive à calculer le nombre de manière qu'il y a de ranger les valeurs et donc le nombre de tableaux standard de young qu'il existe pour une certaine forme. La formule s'écrit

$$f^\lambda = \frac{n!}{\prod h_\lambda(i,j)},$$

où f^λ désigne le nombre total de SYT d'une certaine forme, $h_\lambda(i,j)$ le produit des "hooks" d'un tableau de Young et n le nombre de cases. Il y a un "hook" par cellule et il est égale à la somme du total des cellules qui suivent en colonne et en ligne en ne comptant qu'une seule fois la cellule de l'origine. On note $h_{i,j}$ le "hook" de la cellule $c_{i,j}$ et on a donc $h_{i,j} = card(c_{i<n+1,j} + card(c_{i,j<\lambda_i+1}) + 1$

Image des hooks de toutes les cases d'un SYT.

7	4	3	1
5	2	1	
2			
1			

On déduit que le nombre de tableaux de SYT de cette forme est de :

$$\frac{9!}{1*1*1*2*2*3*4*5*7}$$

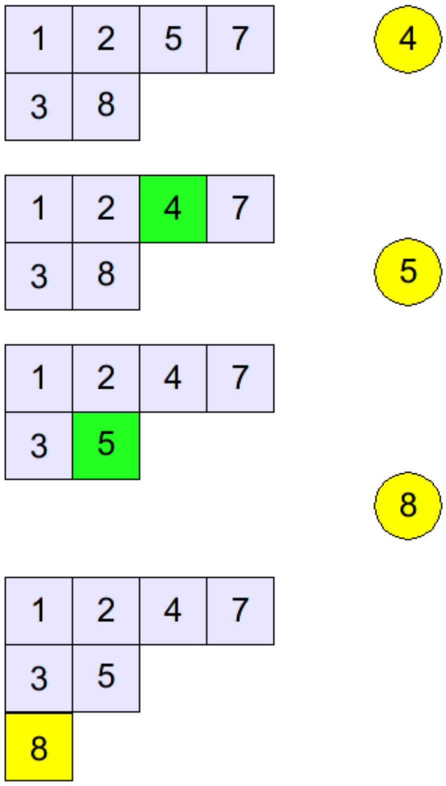
La formule qu'on fait correspondre à un tableau de la forme :

$$X * n - 1$$
$$X * n - 2$$
$$...$$
$$X * 2$$
$$X$$

où X représente une case du tableau.
On se demande donc s'il y a une bijection entre les manières d'effectuer les permutations avec les hirondelles et le nombre de tableaux de young de cette forme, ce qui impliquerait d'avoir un tableau de Young pour chaque série de permutation dans un certain ordre.

C'est là où l'algorithme de Schensted et la correspondance de R.S.K. (Pour Robinson-Schensted-Knuth) entre en jeu :
L'algorithme de Schensted nous permet à partir d'un n-uplet désordonné de trouver un tableau de young de n cases de la forme que l'on veut.
L'algorithme fonctionne de la manière suivante :

1. Soit $i = 1$
2. Soit j le plus petit indice telle que $x < T_{i,j}$ ou, sinon, soit j la longueur de la ligne plus 1.
3. Si la cellule (i, j) est vide dans T, poser $T_{i,j} = x$ et $s = (i, j)$ et terminer.
4. Sinon échanger les valeurs x et $T_{i,j}$, augmenter i de 1 et continuer à l'étape 2.



Pour prouver une bijection entre les suites de permutations et les SYT d'une certaine forme à l'aide de la méthode de la correspondance de RSK on utilise cet algorithme, une matrice $P_{i,j}$ avec i pour les lignes et j pour les colonnes à 2 lignes pour noter l'ordre de nos permutations et leur détails dans un n-uplets Q tel que $P_{1,j}$ nous donne la j-ème valeur permuté et $Q_{P_{1,j}}$ la valeur avec laquelle elle est permutée. On intitialise aussi couple (A, B) de deux SYT.

Dans la matrice P, sur la première ligne on note l'ordre des permutation et ses valeurs vont donc simplement de 1 à n en incrémentant de 1 à chaque fois et sur la deuxième ligne les valeurs qui vont être permutés.
On applique l'algorithme de Schensted dans l'ordre d'indexation des valeurs pour la première ligne de la matrice sur le SYT A et sur le SYT B pour la deuxième ligne. On obtient un couple unique et on peut retrouver l'ordre dans lesquelles les permutations ont été faites en se basant sur la manière dont sont disposer les valeurs dans les cases de A

On imagine un thème où il faut permuter $2, 5, 3, 7, 6, 1, 4$ avec $Q = 3, 6, 4, 2, 3, 2, 1$
Exemple pour la matrice de permutations : $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 5 & 3 & 7 & 6 & 1 & 4 \end{pmatrix}$

On a les SYT B et A qui sont respectivement égale à :

$$\begin{ytableau} 2 \end{ytableau} \to \begin{ytableau} 2 & 5 \end{ytableau} \to \begin{ytableau} 2 & 3 \\ 5 \end{ytableau} \to \begin{ytableau} 2 & 3 & 7 \\ 5 \end{ytableau} \to \begin{ytableau} 2 & 3 & 6 \\ 5 & 7 \end{ytableau} \to \begin{ytableau} 1 & 3 & 6 \\ 2 & 7 \\ 5 \end{ytableau} \to \begin{ytableau} 1 & 3 & 4 \\ 2 & 6 \\ 5 & 7 \end{ytableau}$$

$$\begin{ytableau} 1 \end{ytableau} \to \begin{ytableau} 1 & 2 \end{ytableau} \to \begin{ytableau} 1 & 2 \\ 3 \end{ytableau} \to \begin{ytableau} 1 & 2 & 4 \\ 3 \end{ytableau} \to \begin{ytableau} 1 & 2 & 4 \\ 3 & 5 \end{ytableau} \to \begin{ytableau} 1 & 2 & 4 \\ 3 & 5 \\ 6 \end{ytableau} \to \begin{ytableau} 1 & 2 & 4 \\ 3 & 5 \\ 6 & 7 \end{ytableau}$$

Vu que à tout instant la forme de A est la même que B alors A partir de ce couple il est possible de retrouver la séquence initiale de permutation et de revenir en arrière étape par étape, il suffit de trouver la dernière valeur dans A et si elle est égale à la dernière valeur dans B alors on regarde s'il n'y a pas une valeur au-dessus qui justifierait qu'elle ait été déplacée dans l'algorithme de Schensted. En l'occurence dans l'exemple on regarde d'abord 7 dans A et dans B, on sait que la dernière valeur aussi grande insérée dans B aurait été à la place du 4 pour que le 6 et le 7 soient décalés. Si c'était le 6 ou le 7 qui avait été insérés en dernier alors selon les règles de l'algorithme de Schensted ils auraient bien remplacés le 4. Avec cet algorithme inverse on retombe finalement sur l'unique suite de permutation qui peut nous permettre d'obtenir ce couple (A, B) de SYT.
On en déduit que pour chaque couple (A, B) on a une série de permutations et vice-versa et que l'on est bien dans le cas d'une bijection.

Le nombre de couples (A, B) et égale au nombre de SYT de B car A est toujours le même dans notre cas.

Par la suite, on est assez chanceux car dans notre cas n le nombre de case est toujours égale à $C(2, n)$ où n désigne cette fois le nombre d'hirondelle donc on peut réécrire que le nombre de cases du tableau est égale à $C(2, n)$ ce qui nous donnera des formes de tableaux où les hooks sont réguliers et toujours égales à $1^{n-1} * 3^{n-2} * .. * (2n-5)^2 * (2n-3)^1$

h = n-2 + n-2 +1 = 2n-3	...n-2 lignes	
...		
n-2 colonnes		
X	X	
h = n-3 + n-3 + 1 = 2n-5	.. n-3 lignes	
..n-3 colonnes		
X	h = 2n-5 aussi	..n-3 lignes
X	... n-3 colonnes	

Dans cette image j'ai essayé de calculer les 3 premiers hooks de la première case et de ses deux cases adjaccentes en ligne et en colonne pour un SYT de la même forme que dans le problème.

On remarque rapidement la tendance, on a $1 * (2n-3)$ puis $(2n-5)^2$ jusqu'à 1^{n-1} car il y a une case en coin (dont le hook sera forcément égale à 1) pour chaque n-1 lignes du SYT.

Voilà donc un bref aperçu d'où vient cette formule.

5 Introduction de la Mésange et des espèces d'oiseaux

5.1 Thèmes obtenables

On considère maintenant d'autres transformations pour voir comment les thèmes vont changer.

Étudions un premier cas où on remplace une hirondelle par une mésange.
Lorsque Elaïa donne à l'étape j une graine à cette mésange situé sur le fil i alors la formule qui lui ait changer de fil est la suivante : $u_{i,j+1} = 2u_{i-1,j} + u_{i+1,j} - u_{i,j}$, la formule restant celle précédente pour toutes les hirondelles.

Propriété 7. *La propriété 1 est toujours valable(Donner deux graines au même oiseau revient à la position avant ces deux distributions)*

Démonstration. Supposons qu'à l'étape j, on donne deux fois une graine à la mésange.
On a alors lors de la première distribution :
$u_{i,j+1} = 2u_{i-1,j} + u_{i+1,j} - u_{i,j}$.
On a ensuite lors de la deuxième distribution :
$u_{i,j+2} = 2u_{i-1,j+1} + u_{i+1,j+1} - u_{i,j+1}$
On a donc $u_{i,j+2} = 2u_{i-1,j} + u_{i+1,j} - (2u_{i-1,j} + u_{i+1,j} - u_{i,j}) = u_{i,j}$
Si on donne une graine à l'hirondelle on revient directement à la propriété 1. □

n=2
Premier cas ; Mésange u_1 et Hirondelle en u_2
Si on lui lance une graine elle arrive en 0 ; en position $m = 1$, une mésange se comporte comme une hirondelle.

À partir de là on a un similaire à celui d'une hirondelle pour la mésange en u_1 ;
$(1,1) \xleftrightarrow{1} (0,1)$

$(1,1) \xleftrightarrow{2} (1,0)$

Si on lance une graine à un des deux oiseaux puis à l'autre alors ce dernier se comportera comme dans le cas $n = 1$, car les bordures se comportent par convention comme des oiseaux sur un fil numéro 0 et que dans ce cas n=1 la mésange comme l'hirondelle se comporte de la même manière

$(1,1) \xleftrightarrow{2} (1,0) \xleftrightarrow{1} (-1,0)$

Si on lance au premier oiseau, puis au deuxième puis une nouvelle fois au premier on a ;
$(1,1) \xleftrightarrow{1} (0,1) \xleftrightarrow{2} (0,-1) \xleftrightarrow{1} (-1,-1)$

Les thèmes obtenables sont 1,1 ; 1,0 ; 0,1 ; 0,-1 ; -1,0 ; -1,-1

Deuxième cas ; Mésange u_2 et Hirondelle en u_1
Si on lui lance une graine elle reste en 1 ; elle ne se comporte plus comme une hirondelle.
$(1,1) \xleftrightarrow{2} (1,1)$ en lançant sur le deuxième oiseau (mésange)

$(1,0) \xleftrightarrow{i} (1,2)$ mésange u_2 <=> $1,2$ hirondelle u_1
<=> $-1,0$ hirondelle u_1 => $-1,-2$ mésange u_2 <=> $-1,-2$ hirondelle u_1

Par la suite, on a beau lui lancer encore des graines elle restera en 1.

Les thèmes obtenables avec $n = 2$ sont 1,1 ; 1,0 ; 0,1 ; 0,-1 ; -1,0 ; -1,-1 ; 1,2 ; -1,-2

5.2 Programme Python

J'ai fait un programme Python qui recense tout les thèmes obtenables pour certains paramètre avec plusieurs output et de nombreux détails pour pouvoir correctement conjecturer de notre côté.

```
import queue
predecessors = {}

def throw(positions,bird_char,bird_number,n):
    a,b = bird_char[bird_number]
    un = positions[bird_number]
    if bird_number == 0:
        positions[bird_number] = b*positions[bird_number+1] - un
    elif bird_number == n-1:
        positions[bird_number] = a*positions[bird_number-1] - un
    else:
        positions[bird_number] = a*positions[bird_number-1] + \
                                 b*positions[bird_number+1] - un
    predecessors[''.join(str(positions))] = bird_number+1
    return positions

def combinations(a,b,n,special_bird):
    bird_char = [(1, 1)] * n
    bird_char[special_bird] = (a, b)
    bird_positions_queue = queue.LifoQueue()
    final = []
    for bird_at_one in range(n):
        starting_position = [0]*n
        starting_position[bird_at_one] = 1
        bird_positions_queue.put(starting_position)
        while not bird_positions_queue.empty():
            new_bird_positions = bird_positions_queue.get()
            for bird_number in range(n):
                potential_new_combination = \
                    throw(new_bird_positions,bird_char,bird_number,n)
                if potential_new_combination not in final:
                    final.append(potential_new_combination.copy())
                    bird_positions_queue.put(potential_new_combination.copy())
    return final

def special_bird(a,b,n):
    foo = []
    for i in range(n):
        foo.append(["L'oiseau spécial de paramètre a,b se trouve en position",
                    i+1])
```

```
        for element in combinations(a,b,n,i):
            foo.append(element)
            foo.append([predecessors[''.join(str(element))]])
    return foo

a = int(input("entrez a [mettez 1 par défaut]"))
b = int(input("entrez b [mettez 1 par défaut]"))
n = int(input("entrez n (supérieur ou égale à 2)"))
print(special_bird(a,b,n))
```

Le programme python note les thèmes obtenus de cette manière :
$U_1, U_2, U_3, ...[g]$ avec g désignant la dernière graine reçue pour obtenir ce thème
On trouve tout les thèmes obtensibles avec $n = 3$ et une mésange :
"L'oiseau spécial de paramètre a,b se trouve en position 1" : -1, 0, 0[3] et 0, 1, 1[2] et 0, 0, 1[1] et 0, 0, -1[2] et 0, -1, -1[1] et 0, -1, 0[1] et -1, -1, 0[3] et -1, 0, 0[1] et 1, 0, 0[3] et 1, 1, 0[3] et 0, 1, 0[3] et -1, -1, -1[2] et 1, 1, 1[3]

"L'oiseau spécial de paramètre a,b se trouve en position 2" : -1, 0, 0[1] et -1, -2, 0[2] et -1, -2, -2[3] et 1, 0, 0[3] et 1, 2, 0[1] et 1, 2, 2[2] et 1, 1, 0[3] et 1, 1, 1[2] et 0, 1, 1[2] et 0, 0, 1[1] et 0, 0, -1[2] et 0, -1, -1[3] et 0, -1, 0[2] et -1, -1, 0[3] et -1, -1, -1[2] et 0, 1, 0[3] et -1, -2, -1[1] et 1, 2, 1[1],

"L'oiseau spécial de paramètre a,b se trouve en position 3" : -1, 0, 0[1] et -1, -1, 0[2] et -1, -1, -2[3] et 0, -1, -2[3] et 0, -1, 0[2] et -1, -2, -2[1] et 0, 1, 0[3] et 0, 1, 2[2] et 1, 1, 2[3] et 1, 2, 2[1] et 1, 1, 0[1] et 1, 0, 0[3] et 0, 0, 1[1] et 0, 1, 1 [3] et 1, 1, 1[3] et 0, 0, -1[2] et 0, -1, -1[1] et -1, -1, -1[3]

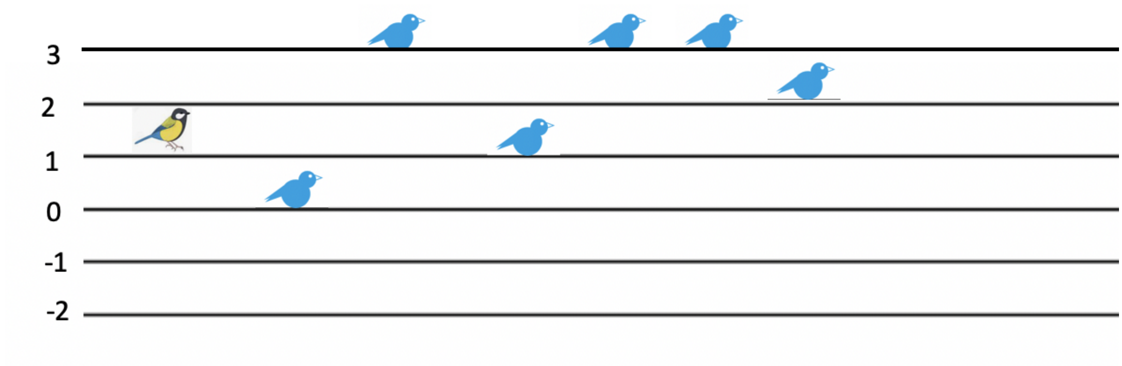
Ici on observe qu'il y a des changements par rapport aux thèmes $n = 3$ obtenus avec exclusivement hirondelles seulement lorsque la mésange est en position 2 c'est à dire entouré d'hirondelles.
On teste avec $n = 4$ sur le programme python et on obtient le même paterne : lorsque la mésange est position 2 et 3 sur 4 c'est à dire entouré elle induira des différences dans les thèmes associés par rapport à ceux obtensibles à l'aide exclusivement d'hirondelles.

Pour certains paramètres notamment lorsque $a > 1$ et $b > 1$ le programme semble trouver une infinité de thèmes. On conjecture pour ces cas là qu'il y a une infinité de thème.

5.3 Sous-cas Triviaux

5.3.1 Mésange en 1

On peut remarquer que si la mésange est le premier oiseau alors, puisque $u(0, j) = 0$ on aura donc le même résultat que pour la transformation d'une hirondelle pour la mésange et donc toutes les transformations précédentes (et donc les résultats obtenus auparavant) sont identiques.



On peut donc réutiliser tout nos résultats précèdents dans ce sous-cas là.

5.3.2 Mésange en n

Un autre sous-cas intéressant est celui où la position de la mésange est en n. Après un certain nombre de graines elle se comporte comme une hirondelle qui serait entouré d'autres hirondelles. On pourrait donc là aussi utilisé les outils développés précèdemment en prenant en compte une hirondelle en position $n+1$ qui serait sur le même câble que la mésange en n qu'on compterait aussi comme une hirondelle. Dans le maximum M cela reviendrait à multiplier par 2 la dernière différence.

5.4 Thèmes définitifs

S'il y a un nombre fini de thèmes associés comme avec le cas de la mésange alors on pourra trouver un M fini tel que le thème sera définitif pour $k \geqslant M$.
Mais si comme dans le cas de $a > 1$ et $b > 1$ on se retrouve avec une infinité de thèmes associées alors pour un nombre de k câbles finis le thème ne sera pas définitif.

5.5 Thèmes Aigus

On pense que dès les thèmes associées sont en nombre limités ils seront donc bordés par un thème aigu et un thème grave.
Parcontre dans tout les cas que nous avons observés les différences de D n'étaient pas conservées et simplement permutées, il est donc impossible d'utiliser un algorithme de tri pour retrouver le thème le plus grave ou le plus aigu.
S'il y a une infinité de thèmes associés alors il ne devrait donc pas y avoir de thèmes plus aigus ou plus graves.

6 Pistes de recherches

Et si on introduisait la génétique :
Selon certaines circonstances, deux oiseaux A et B peuvent se reproduire donnant un œuf qui éclot au bout d'un nombre p de graines donnant un oiseau C, qui a pour fonction :

$$C(x) = \frac{A(x) + B(x)}{2}$$

Et si on pouvait lancer plusieurs graines en même temps.
Et si Elaïa avait une voisine de palier (qui peut aussi donner a manger/chasser des oiseaux/-pièger certains fils/rajouter des fils).

6.1 Conclusion

Lorsqu'on a commencé à étudier le problème j'avais un peu de mal à comprendre tout les enjeux. Mais une fois ce premier obstacle passé, le problème était enfaite entièrement à ma portée et c'est très vite devenu mon préféré. J'ai été agréablement surpris lorsque l'algorithmie a été introduite étant moi même féru de problèmes d'algorithmie. La question 3)D) a été notamment l'une des plus enrichissantes et intéressante. J'ai simplement commencé par un programme Python lorsque j'ai vu que conjecturer à la main devenait long (je suis assez fainéant et astucieux). J'ai fini par étudier et approfondir mes connaissances en combinatoire avec les tableaux de young et la "hook length formula" (j'ai bien aimée comme elle a été prouvée de différentes manières avec différentes approches mêmes fallacieuses - comme celle de Knuth - pour certaines !) mais aussi sur la manière algorithmique de raisonner procéder à une bijection qui m'a vraiment étonné car je ne savais même pas que ce genre de bijection était possible !