Montante descendante

Année 2022 – 2023

Élèves de $3^{ème}$ **:** SHEN-NGUYEN Anne-Lou, XAYGNABOUN-MOUNIER Laurine, NGUYEN Thanh Nhat Nam, Maxime DECK, Lysiane COURTAL, Irène LEMAIRE et Clémentine JAUD

Établissements : Collège Alain-Fournier et Collège Alexander Fleming.

Enseignantes : Florence FERRY et Delphine FILLION.

Chercheurs : Emmanuel KAMMERER, Ecole polytechnique Paris-Saclay et Balthazar FLECHELLES à l'IHES à Bures sur Yvette.

Le sujet

Un nombre pair ($2n$) de joueurs d'échecs sont placés deux par deux sur des tables de 1 à n.

A chaque étape de la montante descendante, le gagnant de la partie dans la table numéro i "monte" à la table $i + 1$ (sauf à la table numéro n où le gagnant reste sur place) tandis que le perdant "descend" à la table i - 1 (sauf à la table numéro 1 où le perdant reste sur place).

On suppose qu'à chaque match, le meilleur joueur gagne et que les joueurs ont des niveaux tous différents.

- A partir de combien d'étapes peut-on être sûr que la table où se situe chaque joueur reflète son niveau ?

- Que se passe-t-il si le perdant de la première table rejoint la dernière et le gagnant de la dernière table rejoint la première ?

I – Compréhension du sujet

Prenons un premier exemple avec 3 tables et 6 joueurs ; le joueur 6 est le plus fort et le joueur 1 le plus faible.

<table>
<tr>
<td>Position de départ

4 | 5

2 | 6

3 | 1

. 5 gagne contre 4, il reste donc à sa place et 4 descend.
. 6 gagne contre 2, il monte donc et 2 descend.
. 3 gagne contre 1, il monte donc et 1 reste à sa place.</td>
<td>Dès la première étape, les joueurs se retrouvent à la place recherchée qui correspond à leur niveau.

5 | 6

3 | 4

1 | 2</td>
</tr>
</table>

Voici un deuxième exemple toujours avec 3 tables et 6 joueurs :

5	**6**	**4**	**6**	**5**	**6**
2	**4**	**3**	**5**	**2**	**4**
1	**3**	**1**	**2**	**1**	**3**

On remarque que les positions des joueurs au départ et celles de la deuxième étape sont les mêmes, et donc nous ne réussirons jamais à obtenir la position recherchée dans ce cas de figure.

Après de nombreux exemples, on s'est aperçu que beaucoup de position initiales ne pouvaient pas aboutir. On s'est demandé quelle condition de départ il fallait sur la position des joueurs pour arriver à la position recherchée.

II – Début de nos recherches

1 - Etude de cas

n est le nombre de tables. On notera E le nombre d'étapes pour arriver à la position recherchée.

a) Si $n = 1$.

Il n'y a qu'un placement pour les deux joueurs qui sont déjà à leur place ; donc : E = 0.

b) Si $n = 2$. Il y a 6 positions de départ.

Cas 1 : tous les joueurs sont bien placés. E = 0. [3 4] [1 2]	Cas 2 : E = 1 [2 4] [3 4] [1 3] [1 2]	Cas 3 : E = 1 [2 3] [3 4] [1 4] [1 2]
Cas 4 : E = 1 [1 4] [3 4] [2 3] [1 2]	Cas 5 : E = 1 [1 3] [3 4] [2 4] [1 2]	Cas 6 : E = 2 [1 2] [2 4] [3 4] [3 4] [1 3] [4 4]

Pour $n = 2$, toutes les positions de départ mènent à la position recherchée en 2 étapes maximum.

Remarques

1 - Les joueurs 1 et $2n$ ne font que monter ou descendre pour rejoindre leur place, il ne font pas "d'aller-retour" puisqu'ils ne croisent jamais plus petit/grand qu'eux.

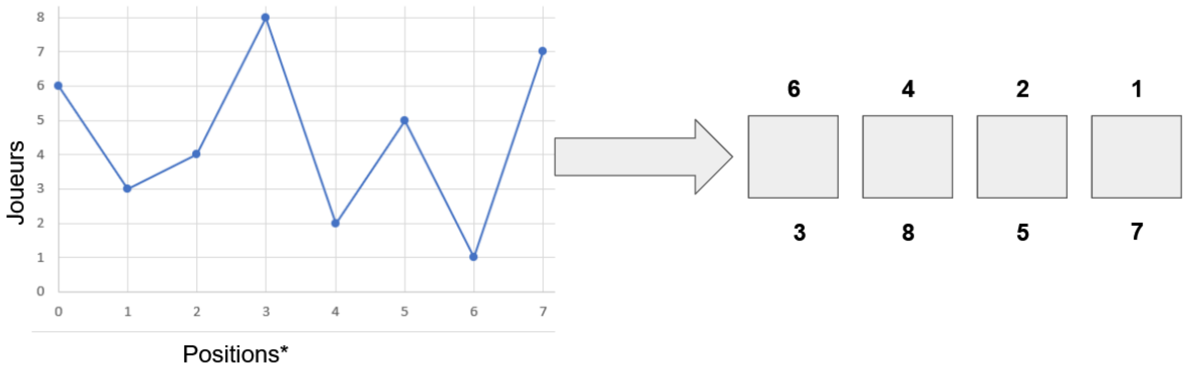
Donc 1 et $2n$ mettent n - 1 étapes au maximum pour rejoindre leur place.

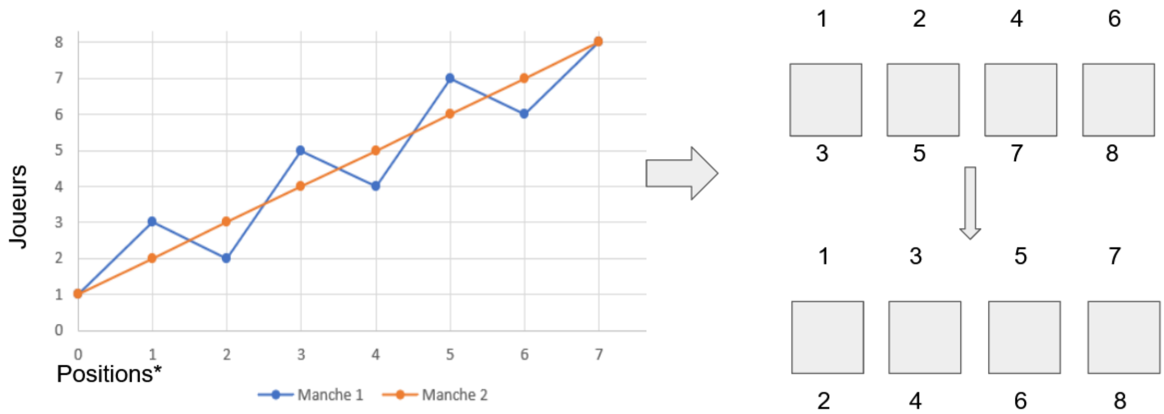
2 - lorsque les joueurs 1 et $2n$ ont rejoint leur place, ils n'en bougent plus.

2 - Simulations graphiques

Les graphiques ont été conçus pour nous permettre de visualiser les combinaisons de manière différente et observer leur évolution sous un nouvel angle. Nous avons conçu deux types de graphiques.

Graphiques statiques

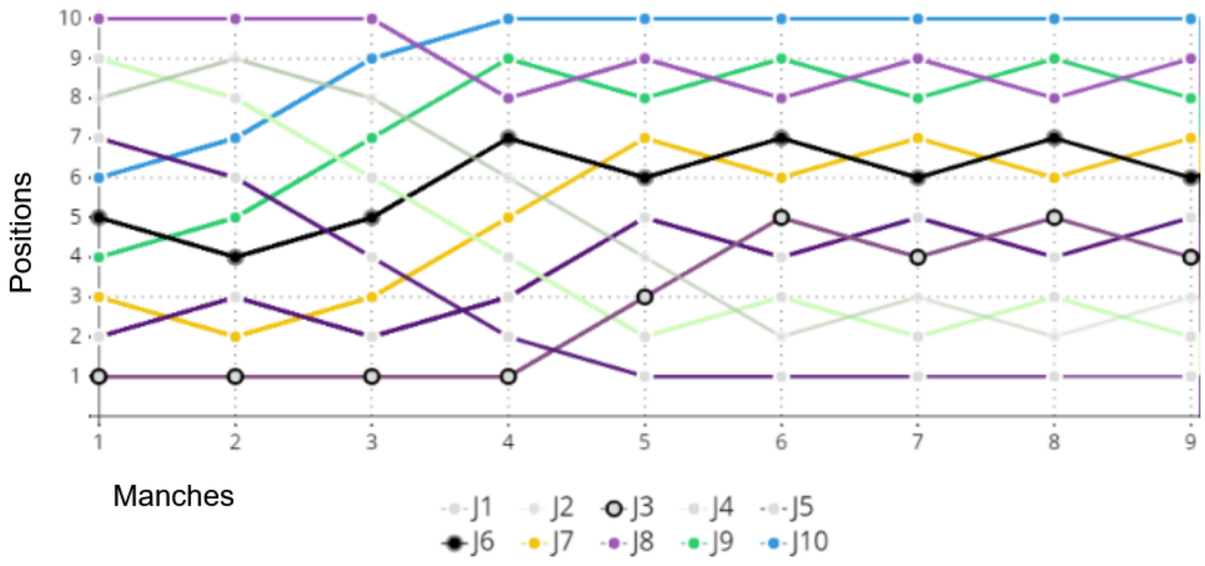




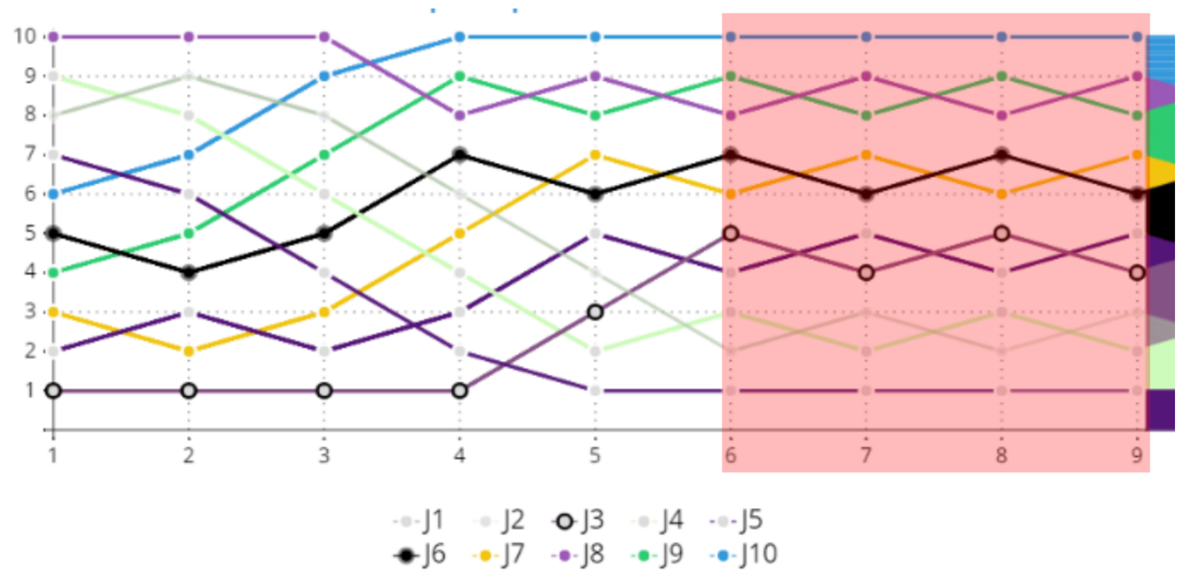
*Positions 8,7 = Table 1;Positions 6,5 = Table 2 ….

L'idée de ce graphique statique est de voir l'évolution des combinaisons en fonction des joueurs. L'axe des abscisses représente les positions des joueurs (une table comporte deux positions). L'axe des ordonnées représente les joueurs. Les courbes représentent les différentes manches.

Graphiques d'évolution



L'idée de ce graphique d'évolution est de représenter graphiquement l'évolution des joueurs. Cela nous a permis d'avoir une vue d'ensemble et de regarder comment les joueurs évoluent. L'axe des abscisses représente les manches et l'axe des ordonnées représente les positions des joueurs. Chaque joueur a une courbe différente représentée par une couleur différente.



Grâce à ce graphique nous avons remarqué qu' à une certaine étape, tous les joueurs restent dans la même zone (représenté ici par la partie rouge).

3 - Recherche informatique

Pour simplifier le processus de calcul des positions, nous avons créé les “MD” qui sont des outils de simulation. Ce logiciel a été développé pour aider les utilisateurs à simuler plus rapidement les combinaisons. En utilisant MD, les utilisateurs auront moins à réfléchir pour plus se concentrer sur la résolution. Grâce à son interface utilisateur intuitive et conviviale, MD permet aux utilisateurs de travailler plus rapidement et plus efficacement.

Il existe plusieurs versions des MD voici une description brève de l’ensemble des versions :

V1 : Le programme prend une combinaison aléatoire pour simuler

V2 : ajout du mode manuel pour que l’utilisateur choisisse une combinaison spécifique

V3 : adaptation pour la deuxième question

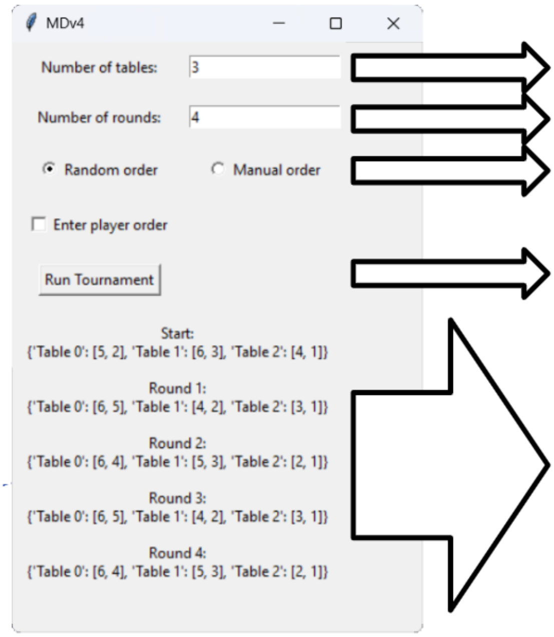
V3.5 : combiner V3 et V2

V4 prototype : V2 en interface graphique

V4 : V3.5 en interface graphique

Présentation de la Version 4 de MD

Mode aléatoire



Entrer le nombre de tables

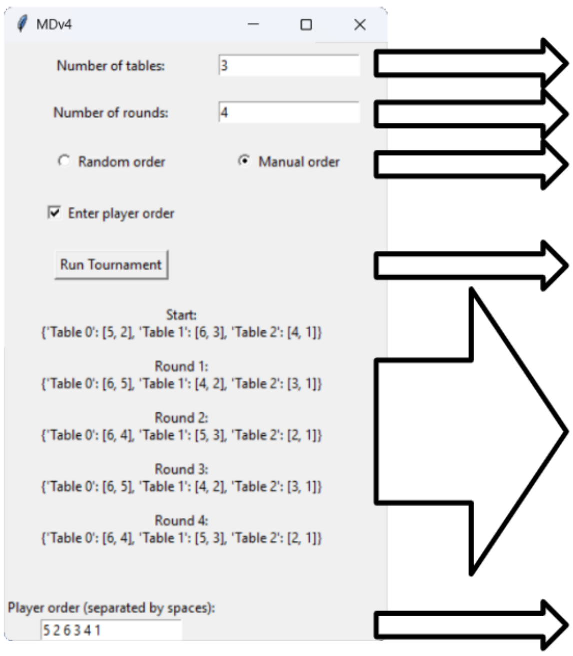
Entrer le nombre de parties à simuler

Choisir entre le mode aléatoire ou mode manuel(ici le mode aléatoire place les joueurs dans un ordre aléatoire).

Button pour commencer la simulation

Sortie

Mode Manuel



Entrer le nombre de tables

Entrer le nombre de parties à simuler

Choisir entre le mode aléatoire ou mode manuel(ici le mode manuel nous laisse placer les joueurs dans l'ordre souhaité).

Button pour commencer la simulation

Sortie

Entrer les joueurs dans l'ordre souhaité

III - Combien y a-t-il de positions de départ possibles ?

Pour n = 3

- Le joueur 1 a trois possibilités de placement, et cinq possibilités d'adversaire => 3×5 positions.
- Il reste 4 numéros de joueurs à placer, et nous utilisons le même raisonnement => $2 \times 3 \times 1$ positions.

Au total, nous obtenons : $3 \times 5 \times 2 \times 3 \times 1$ positions.

Pour n = 4

- Le joueur 1 a quatre possibilités de placement, et sept possibilités d'adversaire => 4×7 positions.
- Il reste 6 joueurs et trois tables, ce qui revient au cas de *n* = 3 vu précédemment => $3 \times 5 \times 2 \times 3 \times 1$ positions.

Au total, nous obtenons : $4 \times 7 \times 3 \times 5 \times 2 \times 3 \times 1$ positions

Généralisons pour un nombre *n* de tables:

- Le joueur 1 a n possibilités de placement, en face de lui, $2n$ - 1 possibilités d'adversaires => $n \times (2n - 1)$ positions.
- Il reste donc le cas précédent => $(n-1)(2n-3)(n-2)(2n-5)\ldots \times 2 \times 3 \times 1$ positions.

Au total, nous obtenons : $n(2n-1)(n-1)(2n-3)(n-2)(2n-5)\ldots \times 2 \times 3 \times 1$

Ce qui donne : $n!\,(2n-1)(2n-3)(2n-5)\ldots \times 3$

IV - Comment prévoir si le résultat recherché sera obtenu ?

Reprenons un exemple qui n'aboutit pas pour essayer de comprendre ce qui empêche que les joueurs rejoignent tous la place qui leur correspond.

Départ		Etape 1		Etape 2		Etape 3		Etape 4	
1	**4**	**3**	**4**	**4**	**6**	**5**	**6**	**4**	**6**
2	**3**	**1**	**6**	**3**	**5**	**2**	**4**	**3**	**5**

5	6		2	5		1	2		1	3		1	2

A la deuxième étape, 1 et 6 sont à leur place et ne bougeront plus. On regarde donc les autres numéros : pour rejoindre leur place, 2 et 3 doivent se déplacer soit 0 fois, soit 2 fois, soit 4 fois, etc, ils doivent effectuer un nombre pair de déplacements ; alors que 4 et 5 doivent se déplacer soit 1 une fois, soit 3 fois, soit 5 fois, etc, ils ont donc un nombre impair de déplacements à effectuer. Quand 2 et 3 seront à leur place, 4 et 5 ne le seront pas et inversement donc cela n'aboutira jamais.

Voici un exemple qui mène à la position recherchée :

4	3		4	2		4	6		5	6
2	1		3	6		2	5		3	4
6	5		1	5		3	1		1	2

A la deuxième étape, 1 et 6 sont bien placés. Tous les autres numéros doivent effectuer un nombre impair de déplacements pour rejoindre leur place, donc cette combinaison de départ aboutit.

Voici encore un exemple qui n'aboutit pas :

3	6		4	6		6	8
2	4		3	8		4	7
1	8		2	7		3	5
5	7		1	5		1	2

A la deuxième étape, 1 et 8 sont bien placés, on regarde donc les autres numéros : 4, 5, 6 et 7 doivent effectuer un nombre impair de déplacements pour aller à leur place, mais 2 et 3 ont, eux, un nombre pair de déplacements à faire, donc cela n'aboutira jamais.

Nos résultats pour 6 joueurs

- Quand 1 joue contre 2, la position recherchée est trouvée seulement si 3 joue contre 4 et 5 contre 6.

- Quand 1 est contre 3, 2 contre 4 et 5 contre 6, cela n'aboutit jamais.

- Quand 1 est contre 4, 2 contre 3, et 5 contre 6, cela n'aboutit jamais.

- Sur les 90 positions de départ possibles, 50 positions n'aboutissent pas.

- La moyenne du nombre d'étapes pour atteindre la position finale est de 1,875 soit environ 2, ce qui revient au nombre d'étapes le plus présent.

- On a trouvé qu'il fallait au maximum 3 étapes pour arriver à la position souhaitée.

V - Résolution de la question 1

Combien d'étapes maximum faudra-t-il pour arriver à la position finale ?

Au lieu de regarder la rangée de table avec les joueurs dans leur ensemble, on traite le problème au cas par cas, c'est-à-dire qu'on ne se concentre plus sur l'évolution de tous les joueurs simultanément mais sur celle d'un seul. Pour répondre à la question, on va chercher le nombre maximal d'étapes que prend un joueur d'un niveau x (le joueur x) pour arriver à sa table i (qui représente le niveau du joueur x).

Pour ce faire, nous avons observé le problème sous un autre angle: le chaperon x (qui représente le joueur d'un niveau x) veut aller chez sa grand-mère qui habite à i (la table représentant le niveau du chaperon). En tant que maître du jeu, nous voulons faire en sorte que le chaperon x prenne le plus de temps possible pour arriver chez sa grand-mère (puisqu'on souhaite trouver le nombre maximal de parties à jouer pour répondre à l'énoncé).

Pour cela, il faudrait 2 choses :

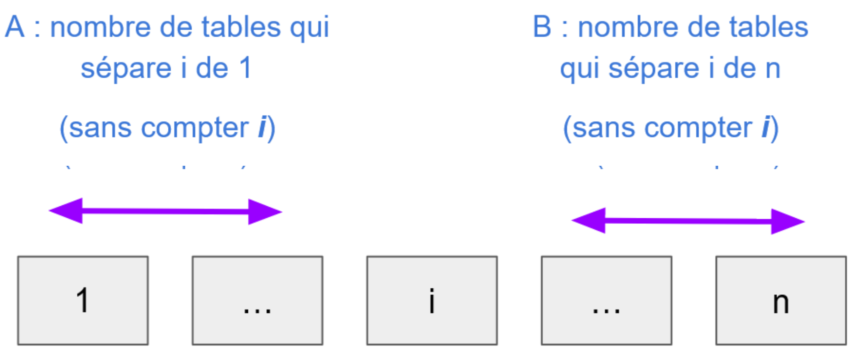
- Il faudrait que le chaperon x commence par jouer à la table la plus éloignée de sa table i.
- Il faudrait que le chaperon x rencontre tous les joueurs qui pourraient le "freiner".

Première partie : Le nombre maximal de table qui sépare la table de départ de x de la table i.

T = Nombre maximal de table qui sépare la table de départ de x de la table i (autrement dit, ce que l'on cherche dans cette partie).

Dans cette partie, on ne prend pas en compte les autres joueurs.

Pour que x prenne le plus de temps possible pour atteindre sa table i, il faut qu'il commence par jouer à un des deux bouts de la rangée de table. Donc soit à la table 1, soit à la table n.
Pour savoir sur quel bout on place x au début, on découpe la rangée de table en deux parties (voir schéma ci-dessous) pour pouvoir comparer ces deux parties.



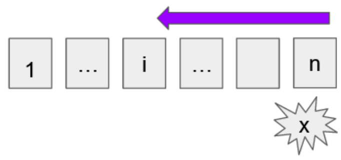
On peut calculer A et B:
A = i - 1
B = n - i

Remarque : A et B sont variables, on ne peut que faire une disjonction de cas ; on différencie trois cas :

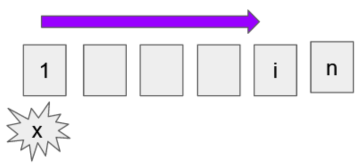
- Dans le premier cas, A < B
- Dans le deuxième cas, A > B
- Dans le troisième cas, A = B

On gardera cette disjonction tout au long du problème, mais nous verrons le troisième cas séparément des deux premiers.

Dans le premier cas, comme la partie la plus longue de la rangée de table est la partie B, on met le joueur x (grâce à nos pouvoirs de maître du jeu) à la table n : c'est la table à laquelle il commence à jouer.



Dans le deuxième cas, comme c'est cette fois la partie A la plus longue, on met le joueur x à la table 1, d'où il sera le plus éloigné de sa table i.



Nous avons maintenant la table de départ de x et le nombre de tables qui le séparent de sa table i. On doit alors prendre en compte les autres joueurs. Nous avons remarqué qu'il y avait certains joueurs qui pouvaient ralentir la course du chaperon vers sa grand-mère: ce sont les loups.

Deuxième partie : Le nombre total de loups (= le nombre total de joueurs qui peuvent freiner x).

J = Le nombre total de loups

On peut différencier trois catégories de joueurs:

- ***ceux qui ont un niveau plus faible que x (qui font gagner le joueur x) (en bleu): il y en a un total de x-1 ;***
- ***ceux qui ont un niveau plus élevé que x (qui font perdre le joueur x) (en vert): il y en a un total de 2n-x***
- ***le joueur x***

Autrement dit: **bleu** < x < **vert**

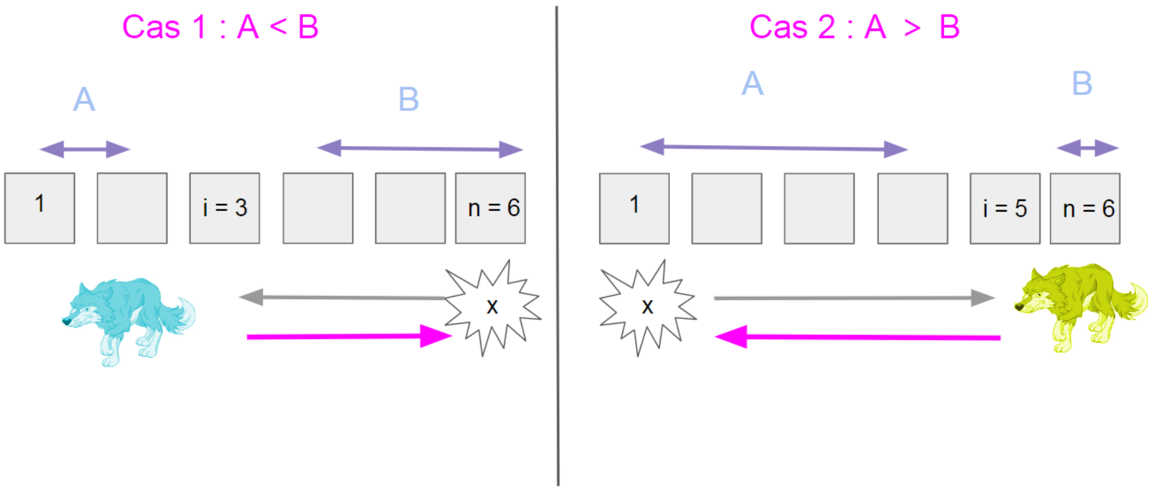
Chaque joueur (à part le joueur x lui-même) peut aider ou ralentir le chaperon x dans sa progression, selon sa catégorie. Ceux qui aident x seront par la suite appelés "chasseurs" et *ceux qui freinent x, les "loups".*

Pour trouver qui sont les loups et qui sont les chasseurs, il faut observer la position dans les deux cas énoncés précédemment :

Dans le premier (A < B), le chaperon x doit perdre pour atteindre sa table i. Les joueurs qui le freinent sont donc ceux qui le font gagner (autrement dit, ce sont les joueurs qui perdent contre x, qui ont un niveau plus faible que lui). **Les loups sont bleus**. **Les chasseurs sont donc verts**.

Dans le deuxième cas (A > B), le chaperon x doit cette fois gagner pour atteindre sa table i. Les joueurs qui le freinent sont donc ceux qui le font perdre (autrement dit, ce sont les joueurs qui gagnent contre x, qui ont un niveau plus élevé que lui). **Les loups sont verts. Les chasseurs sont donc bleus.**

Autrement dit:



Nous savons maintenant que:
- dans le cas 1, les loups sont plus faibles que x et il y en a un total de x-1,
- dans le cas 2, les loups sont plus forts que x et il y en a un total de 2n-x.

Troisième partie : Le meilleur appariement pour les loups

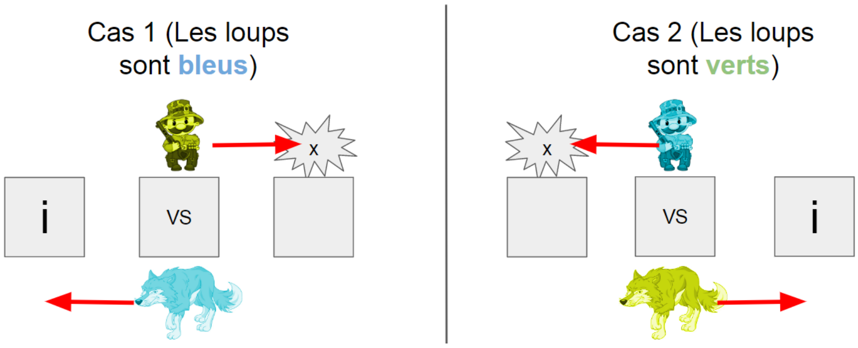
Dans cette partie, on se focalise plus sur les loups de façon individuelle, que sur le chaperon x lui-même.

Tous les loups ne peuvent pas rencontrer x. Nous avons en effet essayé de nombreuses fois de faire rencontrer tous les loups, mais à la suite de ces tests ce défi s'est avéré impossible à réaliser. On s'est donc demandé quel était le meilleur appariement pour qu'un maximum de loups (et peut-être tous les loups) puissent rencontrer le chaperon x.

<u>Les loups ont trois appariements possibles:</u>

- loup VS chaperon x ;
- loup VS autre loup ;
- loup VS joueur qui n'est ni un loup, ni le chaperon, c'est-à-dire, un chasseur.

On ne va pas s'intéresser au premier appariement, puisqu'on connaît déjà le résultat de celui-ci (voir partie 2). Le deuxième appariement pourrait nous intéresser, mais il est assez difficile à étudier, du fait des trop nombreux schémas possibles. On observe donc le plus simple à savoir ce qu'il se passe quand un loup rencontre un chasseur.



A chaque fois, le loup s'éloigne du chaperon x, qui était pourtant son objectif (les loups doivent rencontrer x pour le freiner autant que possible).

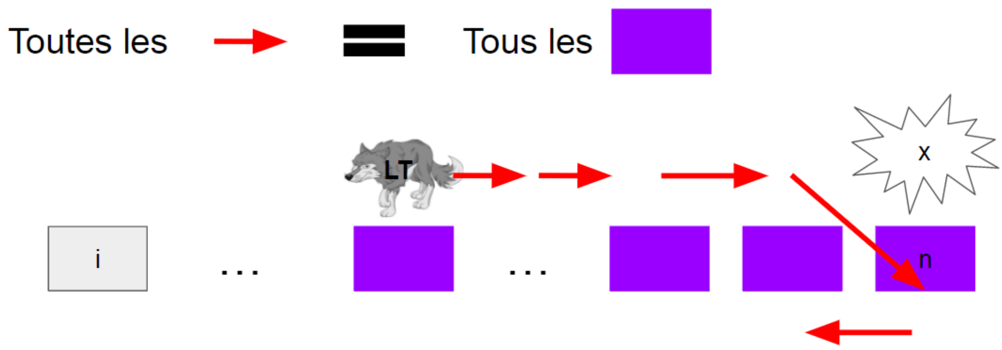
Le meilleur appariement pour les loups est donc un appariement où les loups évitent de jouer au maximum contre les chasseurs en ne jouant qu'entre eux et en étant le plus proche du chaperon x pour pouvoir le retarder dans sa progression (à chaque fois qu'un loup rencontre un chasseur, le loup s'éloigne de x et ne pourra plus revenir).

Nous avons désormais la pire combinaison pour le chaperon x, qui est également la meilleure pour le cruel maître du jeu. Mais il faut encore déterminer le nombre d'étapes durant lesquelles le chaperon x rencontre les loups.

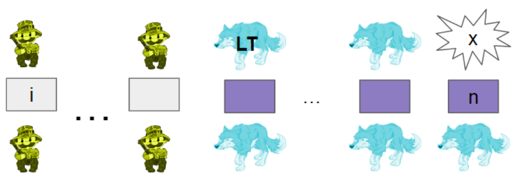
Quatrième partie : Le nombre maximal d'étapes durant lesquelles le chaperon x rencontre les loups.

Il n'existe qu'un seul loup capable de renvoyer tous les autres loups vers les chasseurs: c'est le loup traître (LT). C'est en fait le loup dont le niveau est le plus proche de x (le joueur dont le niveau est x-1 dans le cas 1, ou celui dont le niveau est x+1 dans le cas 2). LT est donc instoppable dans sa course vers le chaperon x. On le place le plus loin de x, mais contre un autre loup, pour ne pas qu'il commence par jouer contre un chasseur et qu'il s'éloigne définitivement de x. Ainsi, LT est le dernier loup qui joue contre le chaperon, c'est le dernier qui retient le chaperon avant qu'il n'aille vers sa grand-mère.

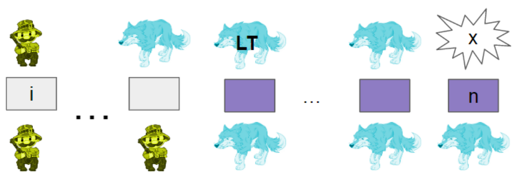
Pour obtenir le nombre d'étapes que le chaperon passe à jouer contre des loups, il suffit de calculer le nombre d'étapes qu'il faut à LT pour atteindre le chaperon, jouer et perdre/gagner contre lui. On se rend compte que ce nombre (que l'on va appeler Jr par la suite) est égal au nombre de tables qui sépare LT du chaperon x. (voir schéma)
Soit Jr = le nombre de tables qui séparent le chaperon x du loup le plus éloigné de x mais qui joue tout de même avec un autre loup



Pour trouver Jr en fonction de J (= nombre total de loup), il suffit d'observer avec un exemple.



Ces deux combinaisons sont quasiment les mêmes, à la seule différence qu'on a ajouté un loup à la deuxième combinaison par rapport à la première. Bien qu'il n'y ait pas le même nombre de loups dans les deux combinaisons, Jr est le même.



Même si faire J/2 pour obtenir Jr fonctionne dans la deuxième possibilité puisque dans cette possibilité, J est pair, cette même technique ne fonctionne pas avec la première possibilité, puisque dans celle-ci, J est impair.

Si on refait quand même cette technique, on obtient un résultat plus petit de 0,5 que le résultat que l'on est censé trouver. Pour corriger cette erreur, on fait un arrondi à l'unité supérieure: $\lceil \frac{J}{2} \rceil$.

Cinquième partie : Calcul du nombre maximal d'étapes que prend le chaperon x pour arriver chez sa grand-mère.

Emax: nombre maximal d’étapes que prend un joueur x pour atteindre sa table i (= table qui représente le niveau du joueur de niveau x).

Emax est égal au nombre maximal d’étapes que le joueur de niveau x prend pour emprunter le chemin qui le mène à sa table i (T) additionné au nombre de fois où il est freiné par des loups (Jr).

Emax = T + Jr

Tableau qui récapitule les valeurs pour calculer Emax:

	CAS 1	CAS 2
T	n - i $= n - \lceil \frac{x}{2} \rceil$ *	i - 1 $= \lceil \frac{x}{2} \rceil - 1$ *
Jr	$\lceil \frac{x-1}{2} \rceil$	$\lceil \frac{2n-x}{2} \rceil$
Emax	$\mathbf{n} - \lceil \frac{x}{2} \rceil + \lceil \frac{x-1}{2} \rceil$	$\lceil \frac{x}{2} \rceil - \mathbf{1} + \lceil \frac{2n-x}{2} \rceil$

* En effet, sur chaque table, il y a deux joueurs: un joueur d'un niveau impair et le joueur du niveau juste au-dessus (donc d'un niveau pair). Comme ces deux joueurs de niveaux différents sont sur la même table, faire $\frac{x}{2}$ pour trouver la table i d'un joueur ne fonctionne qu'avec le joueur dont le niveau est pair. Si on fait la même chose avec le joueur de niveau impair, on trouve le bon résultat diminué d'un demi. Pour quand même trouver le bon résultat, on arrondit à l'unité supérieurs: i = $\lceil \frac{x}{2} \rceil$.
Cela peut s'expliquer car sur la table 1 (dans la combinaison où tous les joueurs sont à la table qui reflète leur niveau), il y a les joueurs 1 et 2 et les joueurs sont rangés par paire. Ainsi, on peut retrouver la table i de n'importe quel joueur grâce à son niveau.
Par exemple, la table du joueur de niveau 353 est la table $\lceil \frac{353}{2} \rceil$, soit la table 177.

- **On calcule Emax pour le CAS 1:** (Emax $= n - \lceil \frac{x}{2} \rceil + \lceil \frac{x-1}{2} \rceil$)

    - Si x est PAIR: on peut écrire x sous la forme 2m (m est un entier).

$\lceil \frac{x}{2} \rceil$	$\lceil \frac{x-1}{2} \rceil$	**Emax**
$= \lceil \frac{2m}{2} \rceil$ $= \mathbf{m}$	$= \lceil \frac{2m-1}{2} \rceil$ $= \lceil m - \frac{1}{2} \rceil$ $= \mathbf{m}$	Emax $= n - \lceil \frac{x}{2} \rceil + \lceil \frac{x-1}{2} \rceil$ Emax $= n - m + m$ **Emax = n**

    - Si x est IMPAIR: on peut écrire x sous la forme 2m-1.

$\lceil \frac{x}{2} \rceil$	$\lceil \frac{x-1}{2} \rceil$	**Emax**

$= \lceil \frac{2m-1}{2} \rceil$ $= \lceil m - \frac{1}{2} \rceil$ $= \mathbf{m}$	$= \lceil \frac{2m-1-1}{2} \rceil$ $= \lceil \frac{2m-2}{2} \rceil$ $= \lceil m - 1 \rceil$ $= \mathbf{m - 1}$	Emax = n - $\lceil \frac{x}{2} \rceil + \lceil \frac{x-1}{2} \rceil$ Emax = n - m + m - 1 **Emax = n - 1**

- **On calcule Emax pour le CAS 2:** (Emax = $\lceil \frac{x}{2} \rceil - 1 + \lceil \frac{2n-x}{2} \rceil$)

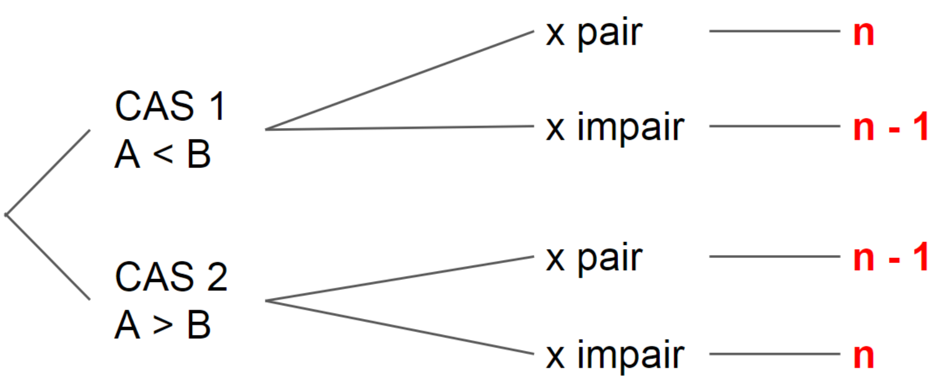
  - Si x est PAIR: on peut écrire x sous la forme 2m.

$\lceil \frac{x}{2} \rceil$	$\lceil \frac{2n-x}{2} \rceil$	**Emax**
$= \lceil \frac{2m}{2} \rceil$ $= \lceil m \rceil$ $= \mathbf{m}$	$= \lceil \frac{2n-2m}{2} \rceil$ $= \lceil n - m \rceil$ $= \mathbf{n - m}$	Emax = $\lceil \frac{x}{2} \rceil - 1 + \lceil \frac{2n-x}{2} \rceil$ Emax = m - 1 + n - m **Emax = n - 1**

  - Si x est IMPAIR: on peut écrire x sous la forme 2m-1.

$\lceil \frac{x}{2} \rceil$	$\lceil \frac{2n-x}{2} \rceil$	**Emax**
$= \lceil \frac{2m-1}{2} \rceil$ $= \lceil m - \frac{1}{2} \rceil$ $= \mathbf{m}$	$= \lceil \frac{2n-(2m-1)}{2} \rceil$ $= \lceil \frac{2n-2m+1}{2} \rceil$ $= \lceil n - m + \frac{1}{2} \rceil$ $= \mathbf{n - m + 1}$	Emax = $\lceil \frac{x}{2} \rceil - 1 + \lceil \frac{2n-x}{2} \rceil$ Emax = m - 1 + n - m + 1 **Emax = n**

Schéma pour résumer les résultats de Emax:



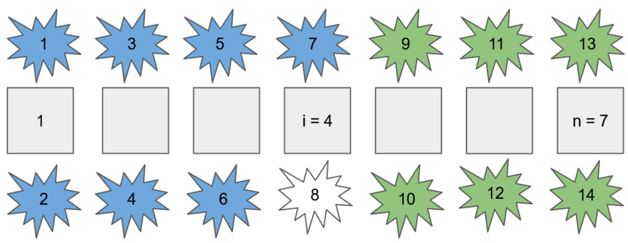
Emax est donc égal à n-1 ou n selon le cas et selon la parité de x. Mais le plus grand résultat trouvé est n.

Cinquième partie: Le troisième cas (A = B)

Ce cas n’arrive que si n est impair, et si on se concentre sur les joueurs de niveaux n ou n+1.

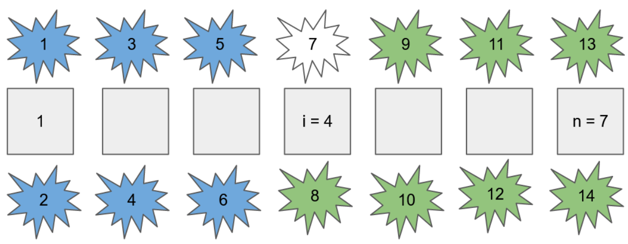
Dans ce cas, comme A = B, on peut commencer par mettre le chaperon x soit à la table 1, soit à la table n pour table de départ. En effet, les deux “chemins” pour arriver à la table i seraient égaux. (Dans l’exemple ci-contre, T est toujours égal à i - 1 = 3, qu’on mette x à la table 1 ou n).

3.a



Le problème qui se pose alors porte sur l’identité des loups.

3.b

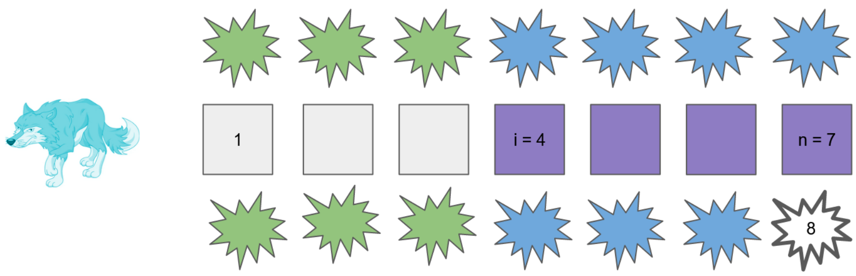


Observons la position où tous les joueurs sont à la table qui reflète leur niveau:

Pour ces deux cas 3.a et 3.b, il y a à chaque fois deux possibilités d'appariement de tous les joueurs: soit on décide que les loups sont bleus, soit on décide qu’ils sont verts.

Les tables en violet sont les tables qui séparent le chaperon x du loup le plus éloigné de x mais qui joue tout de même avec un autre loup (ce nombre de table nous donne Jr).

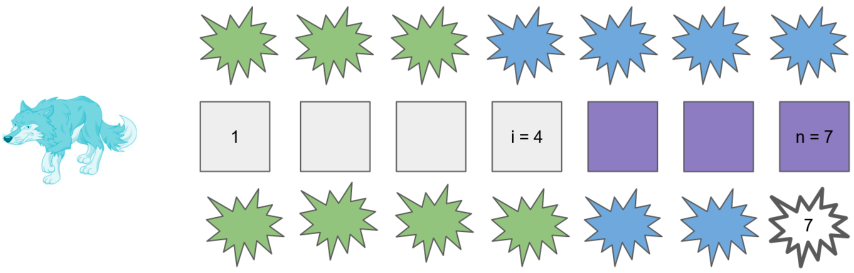
3.a



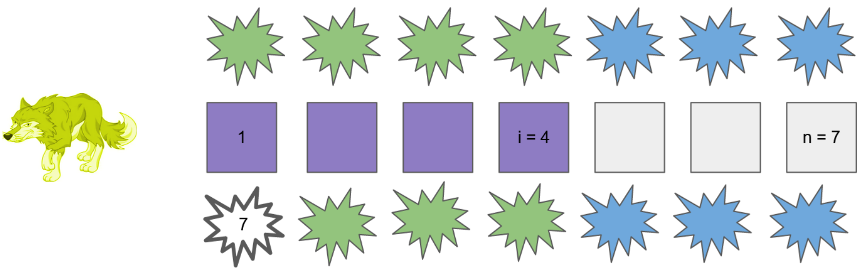
Ici, les loups sont bleus.
Jr = i
(Ici, Jr=4)

Ici, les loups sont verts..
Jr = i - 1
(Ici, Jr=3)

3.b



Ici, les loups sont bleus.
Jr = i - 1
(Ici, Jr=3)



Ici, les loups sont verts..
Jr = i
(Ici, Jr=4)

Comme on veut le nombre <u>maximal</u> d'étapes, on ne prend pas en compte la plus petite valeur de Jr (donc i - 1). On peut maintenant calculer Emax:

Emax = T + Jr
Emax = i - 1 + i
Emax = 2i - 1

Dans ce troisième cas, i a la particularité d'être égal à $\lceil n \div 2 \rceil$ (car n est forcément impair).

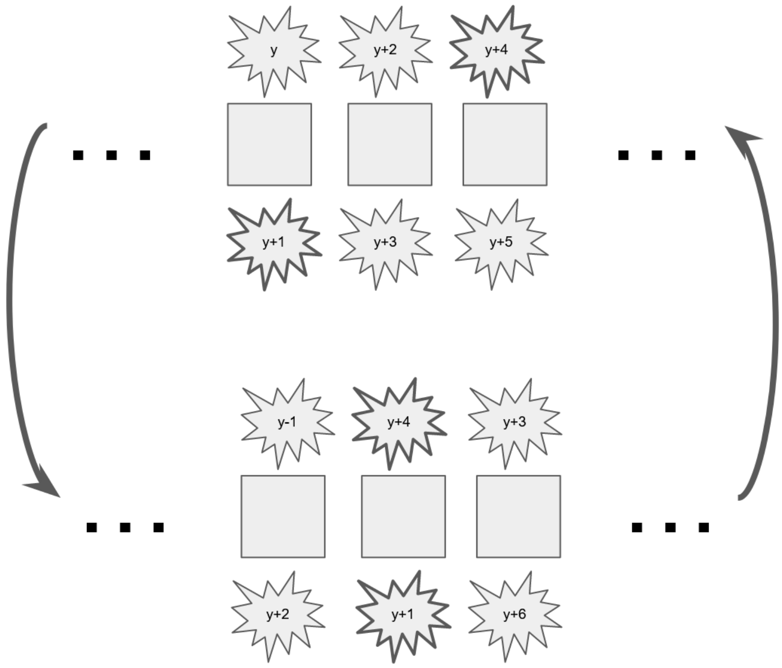
Emax = $2 \times \lceil n \div 2 \rceil - 1$
Emax = n - 1

Dans ce troisième cas, Emax = n - 1.

<u>Conclusion :</u> Le plus grand Emax que nous avons trouvé est donc n.

Remarque : On a remarqué qu'à partir de la combinaison recherchée (= la combinaison dans laquelle chaque joueur est à la table qui représente son niveau au bout de n étapes), si on continue de faire jouer les joueurs, la combinaison recherchée revient après 2 étapes.

Ici, y représente le niveau d'un joueur quelconque.



On observe que les joueurs y+1 et y+4 reviennent à leurs places au bout de 2 étapes.

On peut en conclure que tous les joueurs reviennent à leur place au bout de 2 étapes.

Si on observe la position des joueurs à une étape seulement, on risque de ne pas voir la combinaison recherchée alors qu'elle pourrait avoir été atteinte.

Lorsque la combinaison recherchée est atteinte, on arrive dans un cycle infini où la combinaison recherchée et une autre combinaison.

Il y a plusieurs types de suite de combinaison (qui aboutissent):

- celles qui aboutissent (=qui atteignent la combinaison recherchée) au bout de n étapes;

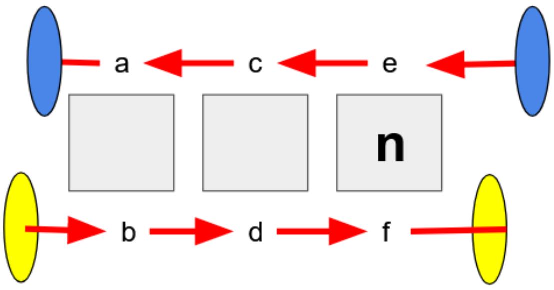
- celles qui aboutissent avant n étapes et qui reviennent à la combinaison recherchée au bout de n-1 étapes (à cause du cycle infini);

- celles qui aboutissent avant n étapes et qui reviennent à la combinaison recherchée au bout de n étapes (à cause du cycle infini).

On doit donc vérifier à deux étapes consécutives au moins pour être sûr d'avoir la combinaison recherchée. Les deux étapes consécutives qui arrivent le plus rapidement et avec lesquelles on est sûr d'avoir la combinaison recherchée sont les étapes n-1 et n.

VII - Résolution de la question 2

Que se passe-t-il si le perdant de la première table (1) rejoint la dernière (n) et le gagnant de la dernière table rejoint la première ?



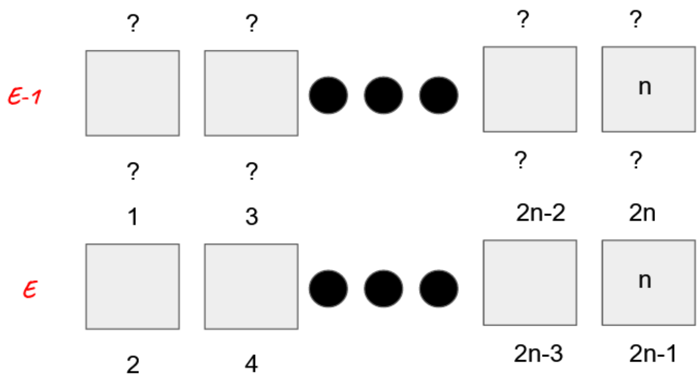
Il faut imaginer qu'il y a un portail après la table n qui relie à un autre portail derrière la table 1. Le gagnant de la table n le traverse de même pour le perdant de la table 1.

Début de la recherche :

Pour commencer, nous avons testé quelques combinaisons. Puis nous avons tenté de trouver un antécédent (une combinaison qui précède une certaine combinaison) de la bonne combinaison. Nous avons remarqué qu'il était impossible d'en trouver.

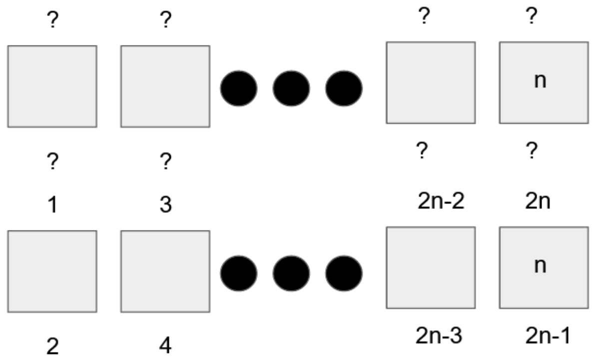
Notre hypothèse est donc la suivante :
Il est impossible de revenir à la bonne combinaison sauf si les joueurs sont bien placés dès le départ.
Autrement dit, $E = 0$ ou $E = \infty$

Démonstration :



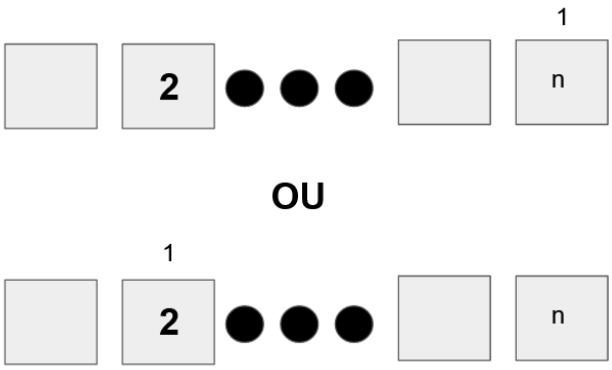
Supposons qu'à l'étape E les joueurs soient bien placés, ce qui signifie forcément qu'à l'étape E - 1 le positionnement des joueurs est favorable.

Condition :



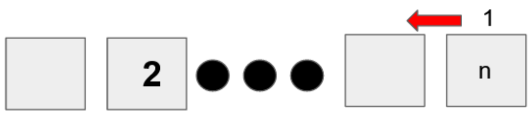
Pour former la bonne combinaison, tout le monde doit arriver à sa table i en bougeant d'une table tous en même temps à un moment donné(car on ne peut pas rester imobile avec ces règles de jeu). La première paire (joueur 1 et joueur 2) doit FORCÉMENT pouvoir se former en UNE SEULE étape

Pour le joueur 1:



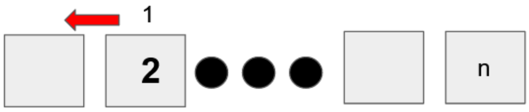
Pour avoir une chance d'atteindre la table 1 en une seule étape, le joueur 1 doit se mettre soit à la table n, soit à la table 2.

Cas 1:



Le joueur 1 étant le plus faible descendra à la table n-1

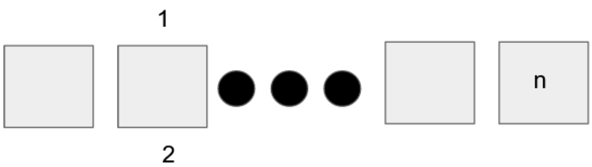
Cas 2:



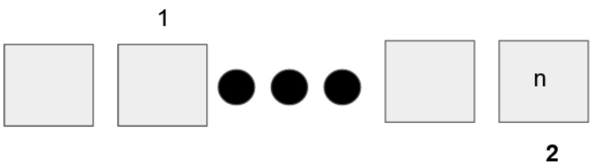
Étant le plus faible, le joueur 1 perdra forcément, ce qui nous arrange car il va forcément aller à la table 1, sa table i

Le joueur 1 ne peut gagner contre personne à la table n, en revanche il perd contre tout le monde.
La table 2 lui est donc la plus favorable.

Pour le joueur 2:

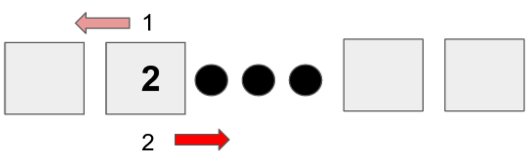


OU



Pour avoir une chance d'atteindre la table 1 en une seule étape, le joueur 2 doit se mettre soit à la table n, soit à la table 2 car sa table i est la table 1.

Cas 1:



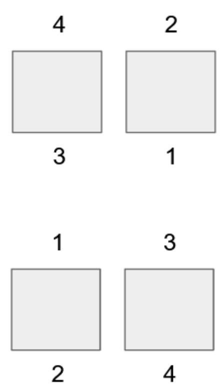
Le joueur 2 gagne contre le joueur 1.

Cas 2:

Le joueur 2 ne peut que gagner contre le joueur 1. Or ce dernier est déjà à la table 2. Le joueur 2 ne peut que perdre donc aller à la table n-1.

Exception:

Pour n = 2, le gagnant et le perdant d'une table vont à la même table. Autrement dit les paires changent de tables.

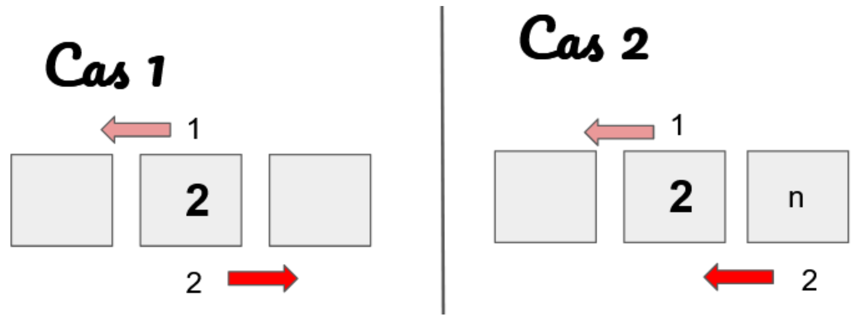


C'est la SEULE combinaison à bien finir avec les nouvelles règles de jeu

Pourquoi est-ce la seule???

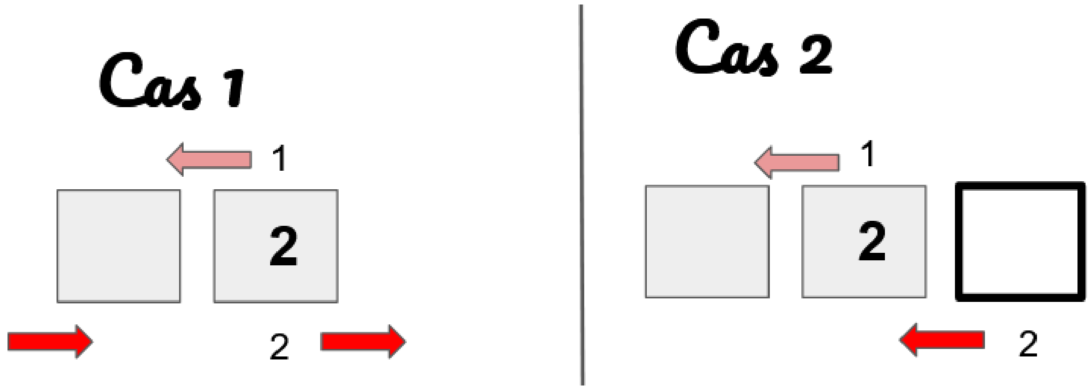
La démonstration marche quand $n \geq 4$ car nous avons au moins 4 tables sur les images.

Cela marche aussi pour $n = 3$



Le joueur 1 se trouve forcément à la table 2 et le joueur 2 se trouve soit à la table n(ici 3) soit à la table 2. Nous voyons que pour les 2 cas la paire 1, 2 ne peut pas se croiser à la table 1 en une étape donc l'hypothèse est vraie pour n = 3.

Pour n = 2 , il n'existe qu'une combinaison capable de finir car comme mentionné précédemment, il n'existe que 2 cas de figures "proches " de la bonne combinaison.



Le cas 1 est possible cependant le cas 2 ne peut pas se réaliser car il n'y a que 2 tables.

=> **Donc c'est la seule combinaison à pouvoir finir avec les nouvelles règles du jeu**