De la couleur dans les idées

Mathematics in Black and White

Bruno Teheux

Unité de recherche en mathématiques
Université du Luxembourg

math.uni.lu/outreach

Une foultitude de problèmes de coloriage mathématiques

Less is more : problèmes à deux couleurs

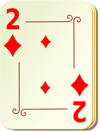
Un approche ludique

Au menu

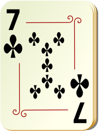
- La couleur ne compte pas… ou bien ?
- L’accord gagnant
- De la suite dans les idées
- Pleins feux !

La couleur ne compte pas... ou bien ?

n_R cartes



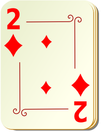
ℓ_N cartes

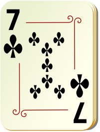


n_r rouges

ℓ_n noires

La couleur ne compte pas… ou bien ?

n_R cartes 

ℓ_N cartes 

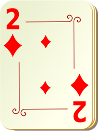
n_r rouges 

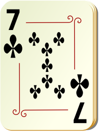
ℓ_n noires 

On procède à deux « double comptages » :

$$\begin{cases} n_R + \ell_N = 26 & \text{cartes faces en l'air} \\ & \end{cases}$$

La couleur ne compte pas… ou bien ?

n_R cartes 

ℓ_N cartes 

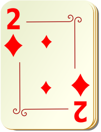
n_r rouges 

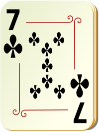
ℓ_n noires 

On procède à deux « double comptages » :

$$\begin{cases} n_R + \ell_N = 26 & \text{cartes faces en l'air} \\ n_R + n_r + (\ell_N - \ell_n) = 26 & \text{cartes rouges} \end{cases}$$

La couleur ne compte pas... ou bien ?

n_R cartes 

ℓ_N cartes 

n_r rouges 

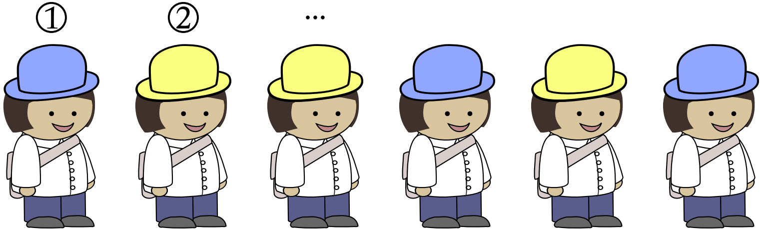
ℓ_n noires 

On procède à deux « double comptages » :

$$\begin{cases} n_R + \ell_N = 26 & \text{cartes faces en l'air} \\ n_R + n_r + (\ell_N - \ell_n) = 26 & \text{cartes rouges} \end{cases}$$

Donc $n_r = \ell_n$

L'accord gagnant

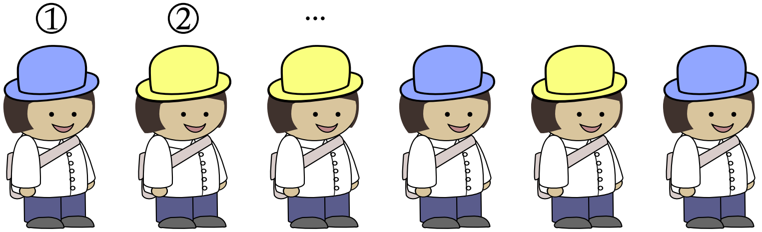


Six personnes alignées ne peuvent tourner le regard

Tour à tour, elles doivent deviner la couleur de leur chapeau

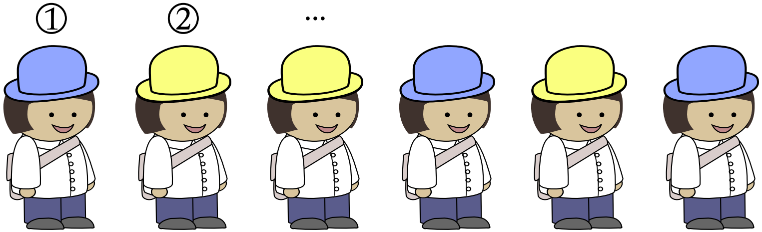
Peuvent-elles suivre une stratégie pour qu'au moins cinq d'entre elles devinent correctement la couleur de leur chapeau ?

L'accord gagnant



Indices :

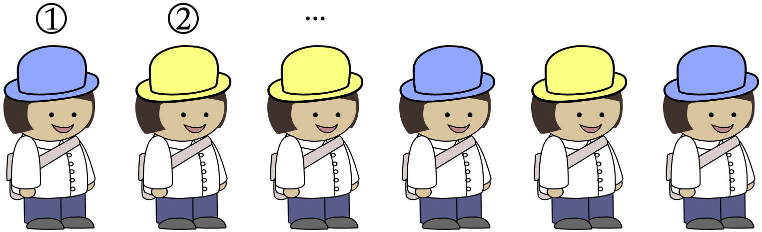
L'accord gagnant



Indices :

La première personne est un « martyr »

L'accord gagnant

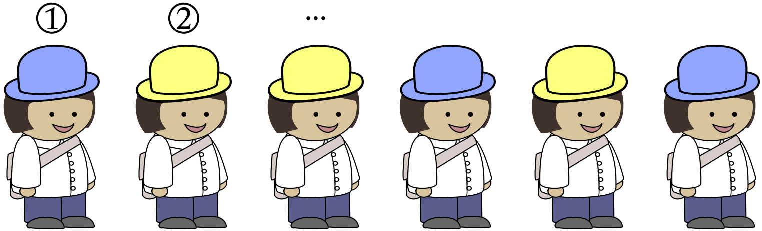


Indices :

La première personne est un « martyr »

La première réponse informe l'*ensemble* des autres joueurs

L'accord gagnant



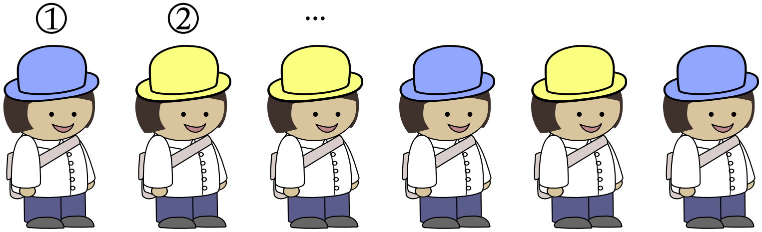
Indices :

La première personne est un « martyr »

La première réponse informe l'*ensemble* des autres joueurs

La première réponse est *binaire*

L'accord gagnant



Indices :

La première personne est un « martyr »

La première réponse informe l'*ensemble* des autres joueurs

La première réponse est *binaire*

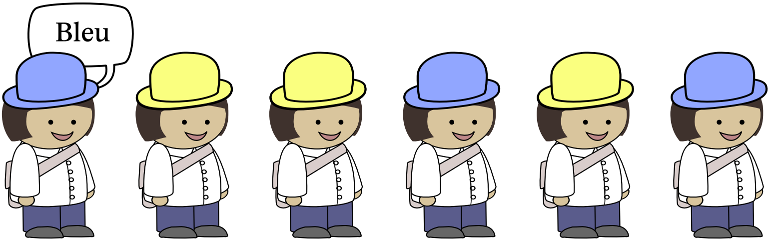
Stratégie :

Le premier réponse spécifie la parité du nombre de chapeaux jaunes

Un exemple de déroulement

Stratégie :

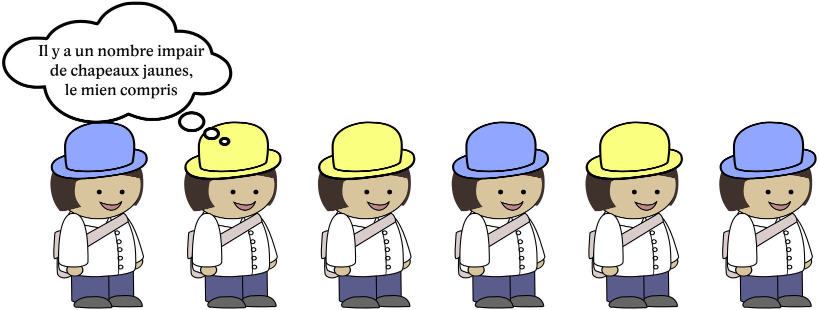
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

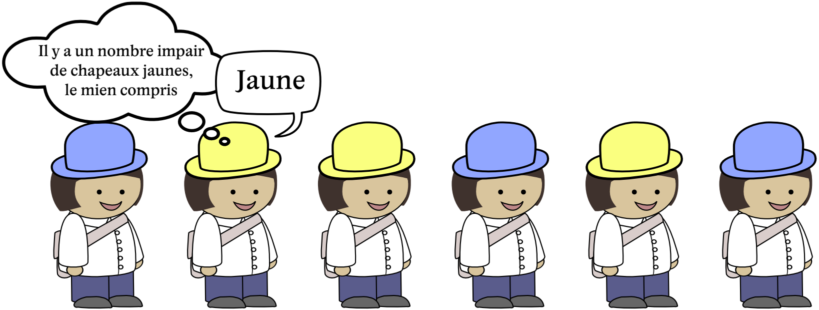
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

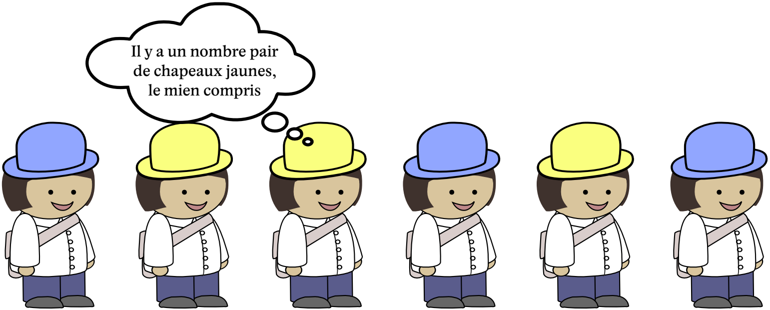
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

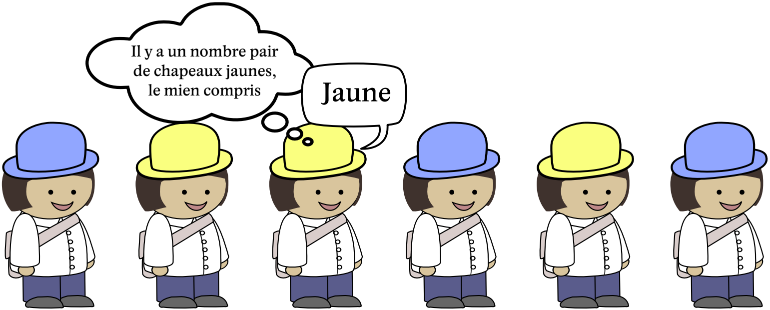
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

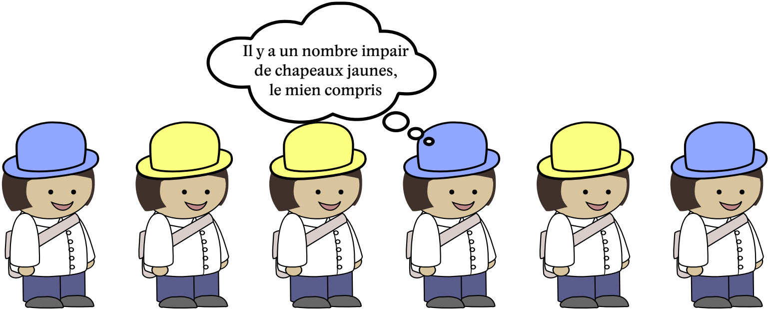
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

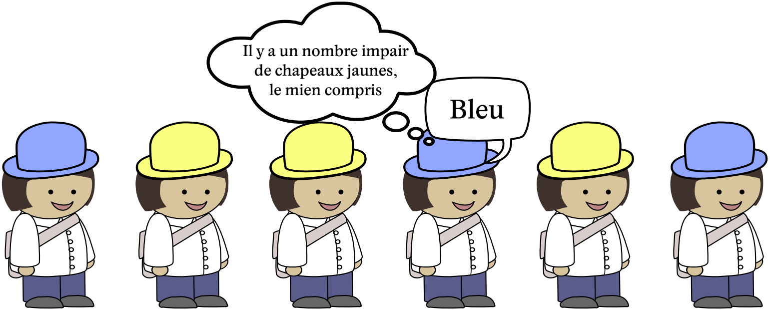
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

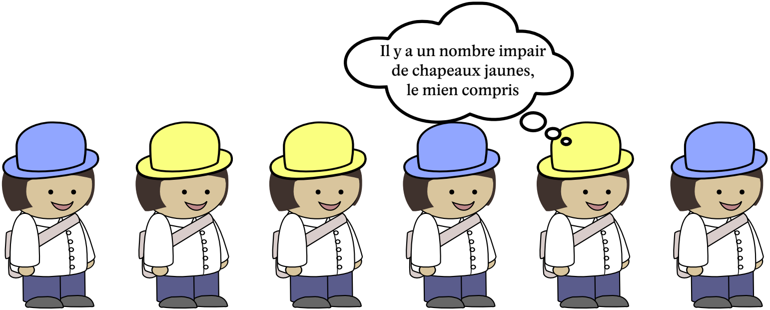
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

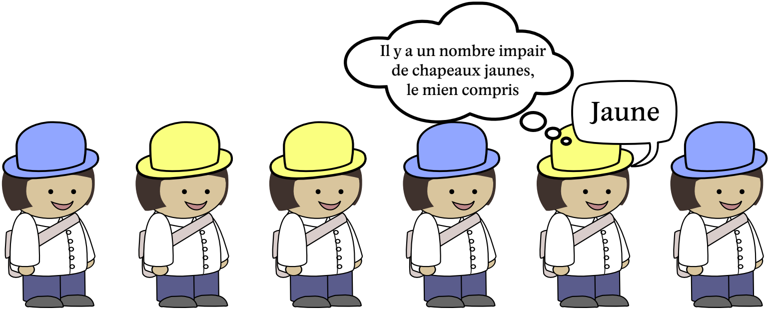
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

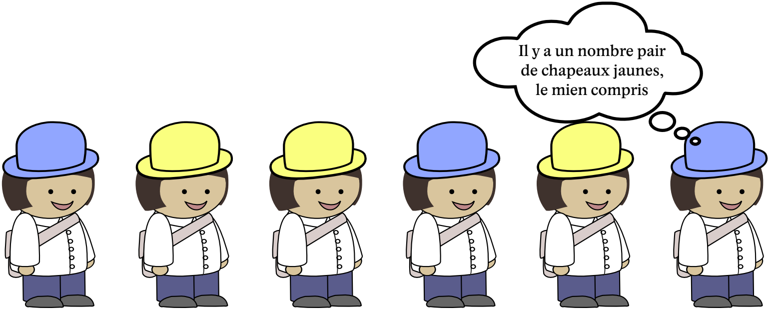
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

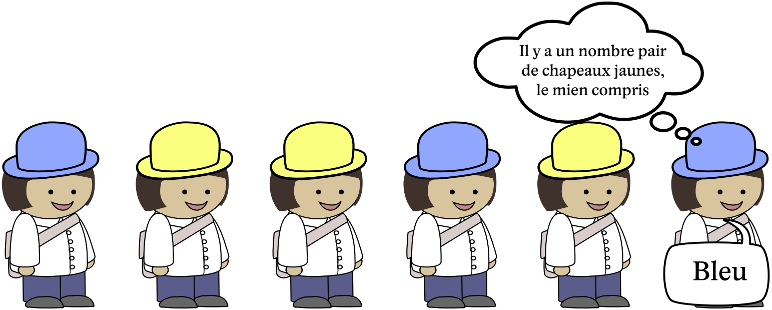
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Un exemple de déroulement

Stratégie :

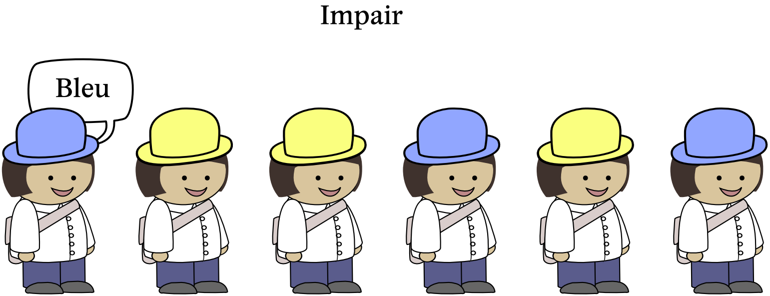
Le premier dit *jaune* s'il voit un nombre *pair* de chapeaux jaunes.



Du point de vue de l'information...

L'information connue de tous se résume par un bit représentant la parité du nombre de chapeaux jaunes restant

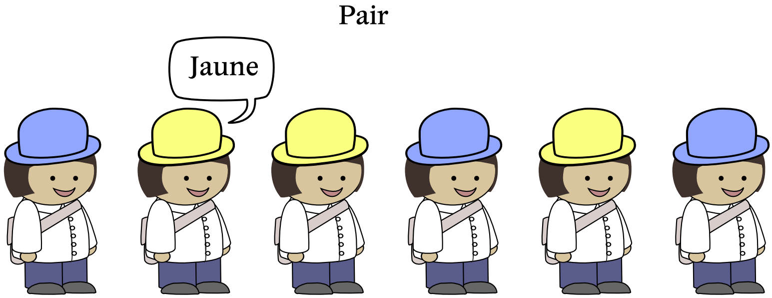
Le bit de parité est mis à jour après chaque réponse



Du point de vue de l'information. . .

L'information connue de tous se résume par un bit représentant la parité du nombre de chapeaux jaunes restant

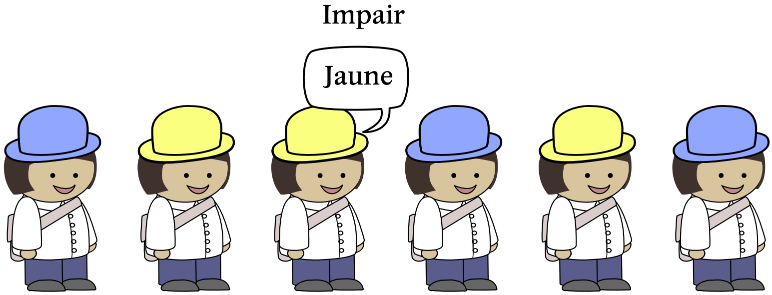
Le bit de parité est mis à jour après chaque réponse



Du point de vue de l'information. . .

L'information connue de tous se résume par un bit représentant la parité du nombre de chapeaux jaunes restant

Le bit de parité est mis à jour après chaque réponse

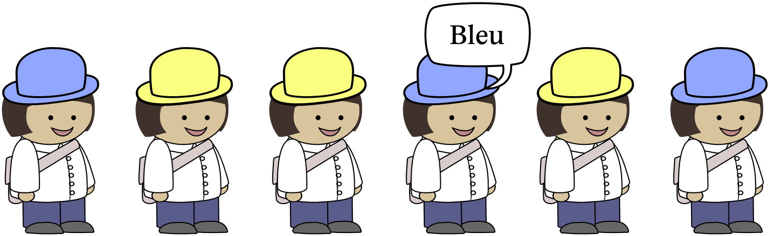


Du point de vue de l'information...

L'information connue de tous se résume par un bit représentant la parité du nombre de chapeaux jaunes restant

Le bit de parité est mis à jour après chaque réponse

Impair

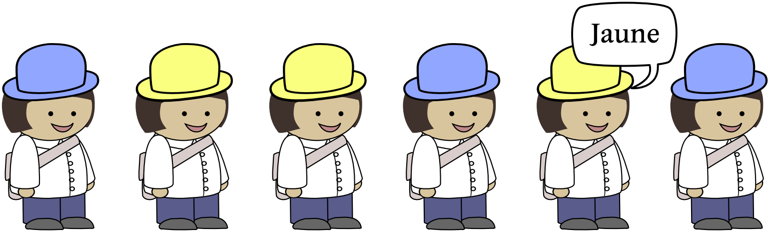


Du point de vue de l'information…

L'information connue de tous se résume par un bit représentant la parité du nombre de chapeaux jaunes restant

Le bit de parité est mis à jour après chaque réponse

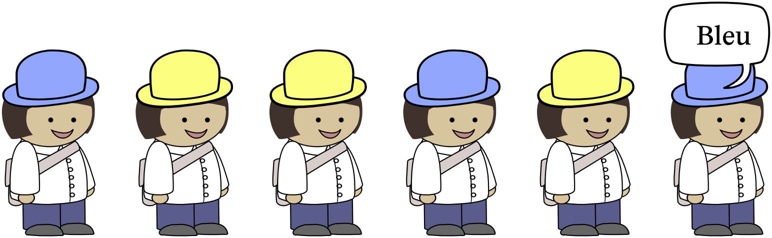
Pair



Du point de vue de l'information…

L'information connue de tous se résume par un bit représentant la parité du nombre de chapeaux jaunes restant

Le bit de parité est mis à jour après chaque réponse



Les informations pertinentes

Il y a 32 cartes dans le jeu

Le jeux est arrangé selon un suite cyclique connue du voyant

Les cinq cartes choisies se succèdent dans la suite

Les informations pertinentes

Il y a 32 cartes dans le jeu

Le jeux est arrangé selon un suite cyclique connue du voyant

Les cinq cartes choisies se succèdent dans la suite

Donc...

Connaître la première carte choisie suffit à déterminer les autres

Comment déterminer la valeur de la première carte ?

Le voyant a comme information

Comment déterminer la valeur de la première carte ?

Le voyant a comme information la suite des couleurs des 5 cartes

Comment déterminer la valeur de la première carte ?

Le voyant a comme information la suite des couleurs des 5 cartes

Il y a

- $2^5 = 32$ suites binaires de longueur 5
- 32 cartes possibles.

Mais. . .

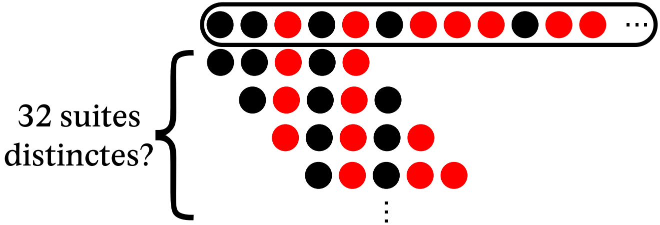
Comment déterminer la valeur de la première carte ?

Le voyant a comme information la suite des couleurs des 5 cartes

Il y a

- $2^5 = 32$ suites binaires de longueur 5
- 32 cartes possibles.

Mais. . .



Comment construire la suite de cartes ?

Existe-t-il une suite binaire

- de longueur 32
- qui contient toute suite binaire de longueur 5 comme sous-suite de 5 éléments consécutifs ?

Comment construire la suite de cartes ?

Existe-t-il une suite binaire

- de longueur 32
- qui contient toute suite binaire de longueur 5 comme sous-suite de 5 éléments consécutifs ?

Oui ! On appelle cela une suite de *de Bruijn*.

Pouvez-vous en construire une ?

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire ● si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire ●

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire ● si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire ●

● ● ● ● ●

Construire une suite de de Bruijn

Algorithme (« $\bullet$ de préférence »)

1. Écrire $\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}$
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire $\bullet$ si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire $\textcolor{red}{\bullet}$

$\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \bullet$

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
    - écrire ● si cela ne crée pas une suite de longueur 5 en double
    - sinon écrire ●

● ● ● ● ● ● ●

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire ● si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire ●

● ● ● ● ● ● ● ●

Construire une suite de de Bruijn

Algorithme (« $\bullet$ de préférence »)

1. Écrire ${\color{red}\bullet} \; {\color{red}\bullet} \; {\color{red}\bullet} \; {\color{red}\bullet} \; {\color{red}\bullet}$
2. Tant que la longueur de la suite n'a pas atteint 32 :
    - écrire $\bullet$ si cela ne crée pas une suite de longueur 5 en double
    - sinon écrire ${\color{red}\bullet}$

$${\color{red}\bullet} \; {\color{red}\bullet} \; {\color{red}\bullet} \; {\color{red}\bullet} \; {\color{red}\bullet} \; \bullet \; \bullet \; \bullet \; \bullet$$

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire ● si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire ●

● ● ● ● ● ● ● ● ● ●

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire ● si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire ●

● ● ● ● ● ● ● ● ● ● ●

Construire une suite de de Bruijn

Algorithme (« $\bullet$ de préférence »)

1. Écrire $\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}$
2. Tant que la longueur de la suite n'a pas atteint 32 :
    - écrire $\bullet$ si cela ne crée pas une suite de longueur 5 en double
    - sinon écrire $\textcolor{red}{\bullet}$

$$\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \textcolor{red}{\bullet}\ \bullet$$

Construire une suite de de Bruijn

Algorithme (« ⚫ de préférence »)

1. Écrire 🔴 🔴 🔴 🔴 🔴
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire ⚫ si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire 🔴

🔴 🔴 🔴 🔴 🔴 ⚫ ⚫ ⚫ ⚫ ⚫ 🔴 ⚫ ⚫

Construire une suite de de Bruijn

Algorithme (« $\bullet$ de préférence »)

1. Écrire $\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}$
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire $\bullet$ si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire $\textcolor{red}{\bullet}$

$$\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \textcolor{red}{\bullet}\ \bullet\ \bullet\ \bullet$$

Construire une suite de de Bruijn

Algorithme (« $\bullet$ de préférence »)

1. Écrire $\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}$
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire $\bullet$ si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire $\textcolor{red}{\bullet}$

$$\textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \textcolor{red}{\bullet}\ \bullet\ \bullet\ \bullet\ \bullet\ \bullet\ \textcolor{red}{\bullet}\ \bullet\ \bullet\ \bullet\ \textcolor{red}{\bullet}$$

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

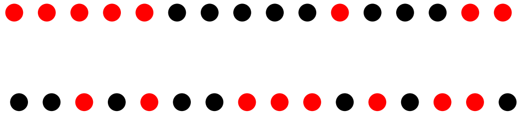
1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
    - écrire ● si cela ne crée pas une suite de longueur 5 en double
    - sinon écrire ●

● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

Construire une suite de de Bruijn

Algorithme (« ● de préférence »)

1. Écrire ● ● ● ● ●
2. Tant que la longueur de la suite n'a pas atteint 32 :
   - écrire ● si cela ne crée pas une suite de longueur 5 en double
   - sinon écrire ●



La suite que j'ai utilisée...

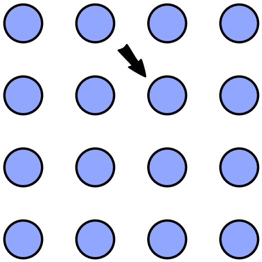
$\clubsuit 2, \clubsuit 3, \clubsuit 6, \spadesuit 3, \diamondsuit 6, \spadesuit 4, \diamondsuit 8, \spadesuit 7, \heartsuit 6, \heartsuit 4, \diamondsuit 7, \spadesuit 5, \heartsuit A, \diamondsuit 2, \clubsuit 4, \clubsuit 8,$
$\spadesuit 8, \heartsuit 8, \heartsuit 7, \heartsuit 5, \heartsuit 2, \diamondsuit 4, \clubsuit 7, \spadesuit 6, \heartsuit 3, \diamondsuit 5, \spadesuit 2, \diamondsuit 3, \clubsuit 5, \spadesuit A, \diamondsuit A, \clubsuit A$

Pleins feux !

Un réseau de jetons bicolores (bleu, jaune)

Au départ, ils sont tous orientés face bleue en l'air

Lorsqu'on retourne un jeton, on doit aussi retourner ses voisins directs sur la même ligne et la même colonne

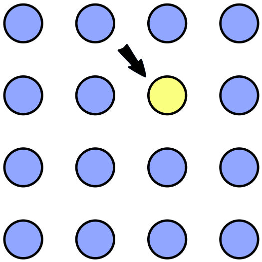


Pleins feux !

Un réseau de jetons bicolores (bleu, jaune)

Au départ, ils sont tous orientés face bleue en l'air

Lorsqu'on retourne un jeton, on doit aussi retourner ses voisins directs sur la même ligne et la même colonne

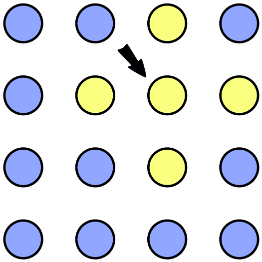


Pleins feux !

Un réseau de jetons bicolores (bleu, jaune)

Au départ, ils sont tous orientés face bleue en l'air

Lorsqu'on retourne un jeton, on doit aussi retourner ses voisins directs sur la même ligne et la même colonne

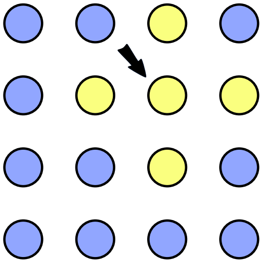


Pleins feux !

Un réseau de jetons bicolores (bleu, jaune)

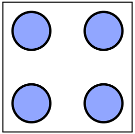
Au départ, ils sont tous orientés face bleue en l'air

Lorsqu'on retourne un jeton, on doit aussi retourner ses voisins directs sur la même ligne et la même colonne



Objectif : configuration où tous les jetons sont face jaunes en l'air

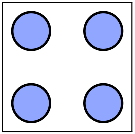
Expérimentons...



L'ordre dans lequel on procède a-t-il de l'importance ?

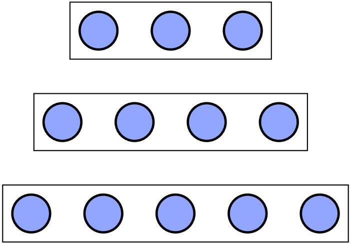
Est-ce utile d'activer deux fois le même jeton ?

Expérimentons. . .



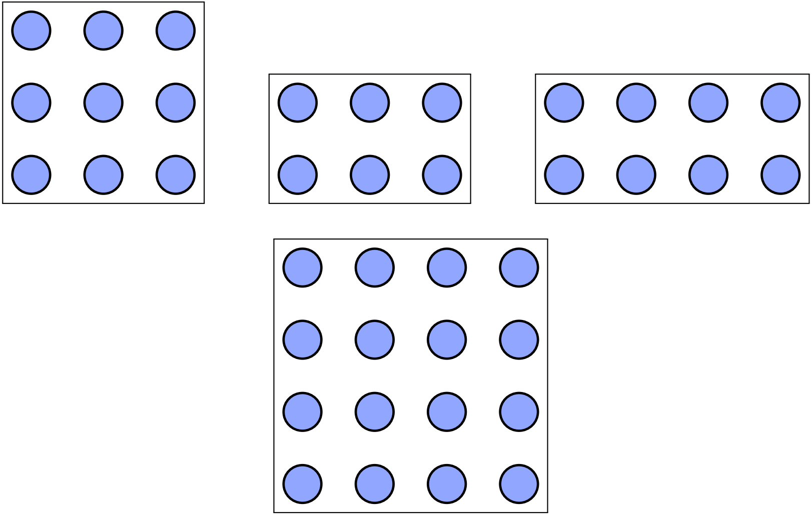
L'ordre dans lequel on procède a-t-il de l'importance ?

Est-ce utile d'activer deux fois le même jeton ?



. . .

Allons plus loin . . .



Un outil pour les résoudre tous

Utilisons l'arithmétique modulo 2 sur $\{0, 1\}$.

On représente l'état du i^{e} jeton par $x_i \in \{0, 1\}$ tel que

- $x_i = 0$ si le bouton i est bleu (éteint)
- $x_i = 1$ si le bouton i est jaune (allumé)

Activer un jeton revient à ajouter 1 à son état et celui de ses voisins.

Un outil pour les résoudre tous

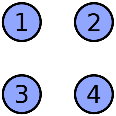
Utilisons l'arithmétique modulo 2 sur $\{0, 1\}$.

On représente l'état du i^{e} jeton par $x_i \in \{0, 1\}$ tel que

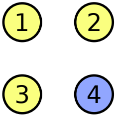
- $x_i = 0$ si le bouton i est bleu (éteint)
- $x_i = 1$ si le bouton i est jaune (allumé)

Activer un jeton revient à ajouter 1 à son état et celui de ses voisins.

Activation du jeton 1



$x_1 = 0,\ x_2 = 0,\ x_3 = 0,\ x_4 = 0$



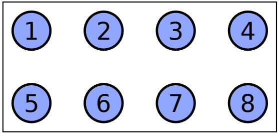
$x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0$

« Pleins feux » est un système linéaire

État initial : chaque jeton est à l'état 0

État final : chaque jeton est à l'état 1

Convention : $a_i = 1$ signifie qu'on active le bouton i.

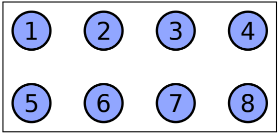


« Pleins feux » est un système linéaire

État initial : chaque jeton est à l'état 0

État final : chaque jeton est à l'état 1

Convention : $a_i = 1$ signifie qu'on active le bouton i.



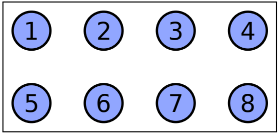
$$\begin{cases} 1 = a_1 + a_2 + a_5 \end{cases}$$

« Pleins feux » est un système linéaire

État initial : chaque jeton est à l'état 0

État final : chaque jeton est à l'état 1

Convention : $a_i = 1$ signifie qu'on active le bouton i.



$$\begin{cases} 1 & = a_1 + a_2 + a_5 \\ 1 & = a_1 + a_2 + a_3 + a_6 \end{cases}$$

« Pleins feux » est un système linéaire

État initial : chaque jeton est à l'état 0

État final : chaque jeton est à l'état 1

Convention : $a_i = 1$ signifie qu'on active le bouton i.

1	2	3	4
5	6	7	8

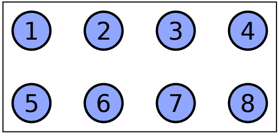
$$\begin{cases} 1 &= a_1 + a_2 + a_5 \\ 1 &= a_1 + a_2 + a_3 + a_6 \\ 1 &= a_2 + a_3 + a_4 + a_7 \\ & \\ & \\ & \\ & \end{cases}$$

« Pleins feux » est un système linéaire

État initial : chaque jeton est à l'état 0

État final : chaque jeton est à l'état 1

Convention : $a_i = 1$ signifie qu'on active le bouton i.



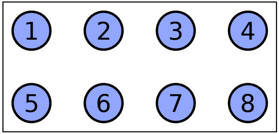
$$\begin{cases} 1 &= a_1 + a_2 + a_5 \\ 1 &= a_1 + a_2 + a_3 + a_6 \\ 1 &= a_2 + a_3 + a_4 + a_7 \\ 1 &= a_3 + a_4 + a_8 \\ 1 &= a_1 + a_5 + a_6 \\ 1 &= a_2 + a_5 + a_6 + a_7 \\ 1 &= a_3 + a_6 + a_7 + a_8 \\ 1 &= a_4 + a_7 + a_8 \end{cases}$$

« Pleins feux » est un système linéaire

État initial : chaque jeton est à l'état 0

État final : chaque jeton est à l'état 1

Convention : $a_i = 1$ signifie qu'on active le bouton i.



$$\begin{cases} 1 &= a_1 + a_2 + a_5 \\ 1 &= a_1 + a_2 + a_3 + a_6 \\ 1 &= a_2 + a_3 + a_4 + a_7 \\ 1 &= a_3 + a_4 + a_8 \\ 1 &= a_1 + a_5 + a_6 \\ 1 &= a_2 + a_5 + a_6 + a_7 \\ 1 &= a_3 + a_6 + a_7 + a_8 \\ 1 &= a_4 + a_7 + a_8 \end{cases}$$

Solution : $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) = (0, 1, 1, 0, 0, 1, 1, 0)$

math.uni.lu/outreach