L'ordinateur est-il tout puissant ?

Introduction à l'informatique théorique

Bruno Grenet

12 octobre 2011

L'Informatique, kesako ?

- Des jeux vidéo ?
- Navigation internet, mails ?
- Traitement de texte, tableurs ?

L'Informatique, kesako ?

- Des jeux vidéo ?
- Navigation internet, mails ?
- Traitement de texte, tableurs ?
- Programmer un logiciel ?
- Mettre en place un réseau ?

L'Informatique, kesako ?

- Des jeux vidéo ?
- Navigation internet, mails ?
- Traitement de texte, tableurs ?
- Programmer un logiciel ?
- Mettre en place un réseau ?

Au cœur de tout ça...

le **CALCUL** !

(en anglais : *Computer* Science)

L'Informatique, kesako ?

- Des jeux vidéo ?
- Navigation internet, mails ?
- Traitement de texte, tableurs ?
- Programmer un logiciel ?
- Mettre en place un réseau ?

Au cœur de tout ça...

le **CALCUL** !

(en anglais : *Computer* Science)

Question

Qu'est ce que le calcul ?

Du temps d'Euclide

Qu'est-ce qu'un nombre ?

Du temps d'Euclide

Qu'est-ce qu'un nombre ?

Un nombre est une quantité qu'on peut construire à la règle et au compas, à partir d'une quantité unité.

Du temps d'Euclide

Qu'est-ce qu'un nombre ?

Un nombre est une quantité qu'on peut construire à la règle et au compas, à partir d'une quantité unité.

Les quantités suivantes étaient-elles des nombres constructibles pour les Grecs ?

$$1 \quad ; \quad 7 \quad ; \quad 5,5 \quad ; \quad \frac{1}{3} \quad ; \quad \sqrt{3} \quad ; \quad \pi$$

$$1+\sqrt{\frac{7,3}{2\times\sqrt{5}}} \quad ; \quad \sqrt[3]{2}$$

Les nombres constructibles

Les constructibles sont stables par :

- $+$ et $-$ $\rightsquigarrow \mathbb{N}$

Les nombres constructibles

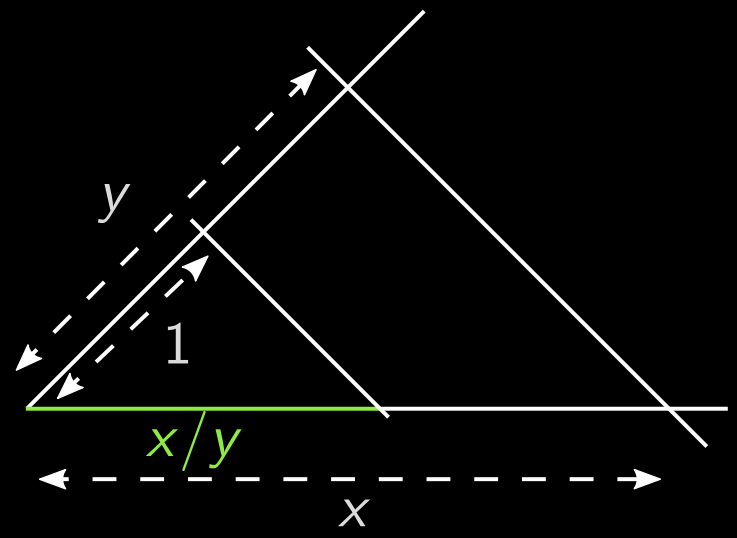
Les constructibles sont stables par :

- $+$ et $-$ $\rightsquigarrow \mathbb{N}$
- $\times$ et $\div$ $\rightsquigarrow \mathbb{Q}$

Les nombres constructibles

Les constructibles sont stables par :

- $+$ et $-$ $\rightsquigarrow \mathbb{N}$
- $\times$ et $\div$ $\rightsquigarrow \mathbb{Q}$



Les nombres constructibles

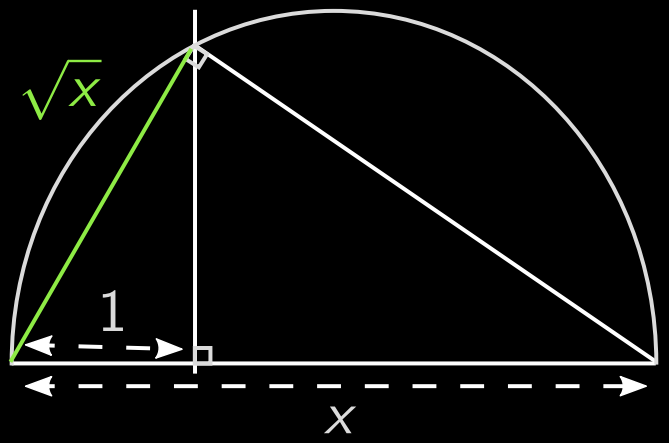
Les constructibles sont stables par :

- $+$ et $-$ $\rightsquigarrow \mathbb{N}$
- $\times$ et $\div$ $\rightsquigarrow \mathbb{Q}$
- $\sqrt{}$ $\rightsquigarrow 1 + \sqrt{\frac{7{,}3}{2 \times \sqrt{5}}}$

Les nombres constructibles

Les constructibles sont stables par :

- $+$ et $-$ $\rightsquigarrow \mathbb{N}$
- $\times$ et $\div$ $\rightsquigarrow \mathbb{Q}$
- $\sqrt{\ }$ $\rightsquigarrow 1+\sqrt{\frac{7{,}3}{2\times\sqrt{5}}}$



Les nombres constructibles

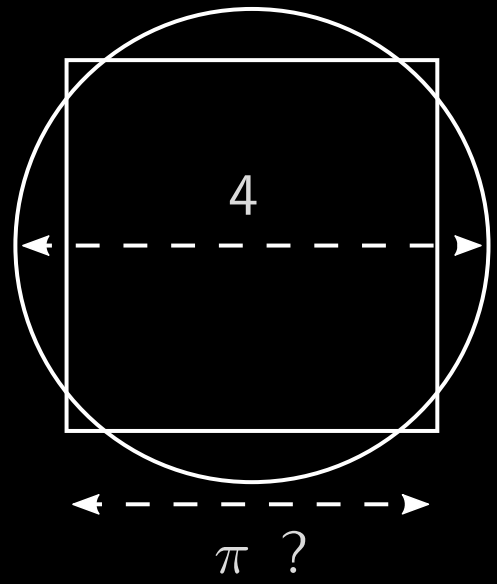
Les constructibles sont stables par :

- $+$ et $-$ $\rightsquigarrow \mathbb{N}$
- $\times$ et $\div$ $\rightsquigarrow \mathbb{Q}$
- $\sqrt{}$ $\rightsquigarrow 1+\sqrt{\frac{7{,}3}{2\times\sqrt{5}}}$
- π et $\sqrt[3]{2}$ ne sont pas constructibles

Les nombres constructibles

Les constructibles sont stables par :

- $+$ et $-$ $\rightsquigarrow \mathbb{N}$
- $\times$ et $\div$ $\rightsquigarrow \mathbb{Q}$
- $\sqrt{\ }$ $\rightsquigarrow 1+\sqrt{\frac{7{,}3}{2\times\sqrt{5}}}$
- π et $\sqrt[3]{2}$ ne sont pas constructibles

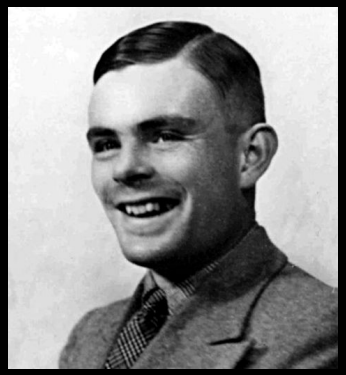


De nos jours

Ça veut dire quoi *calculable* ?

De nos jours

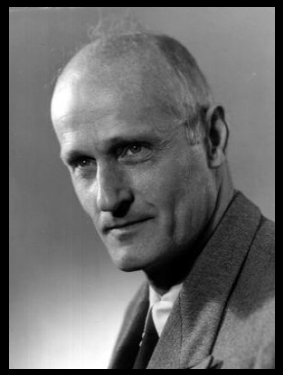
Ça veut dire quoi *calculable* ?



Alan Turing

1936 : 3 réponses

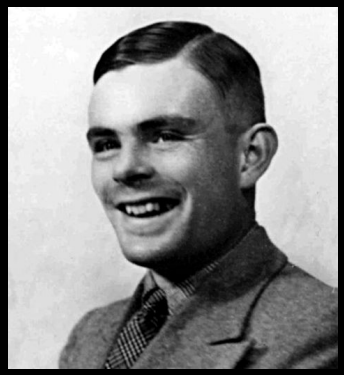
Alonzo Church



Stephen Cole Kleene

De nos jours

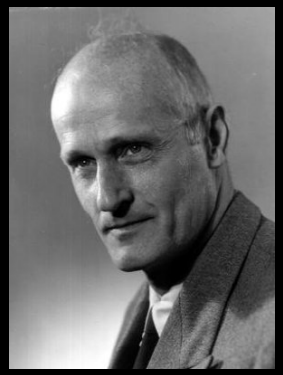
Ça veut dire quoi *calculable* ?



Alan Turing

1936 : 3 réponses

Alonzo Church

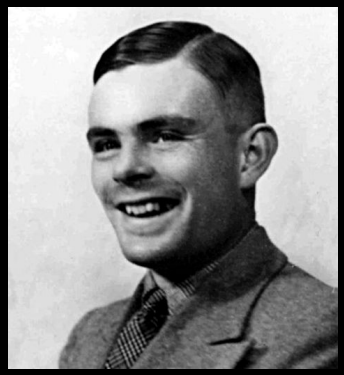


Stephen Cole Kleene

Un problème est *calculable* s'il y a un algorithme qui le résout.

De nos jours

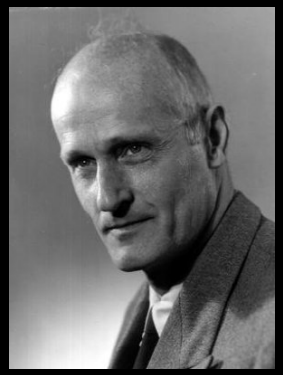
Ça veut dire quoi *calculable* ?



Alan Turing

1936 : 3 réponses

Alonzo Church



Stephen Cole Kleene

Un problème est *calculable* s'il y a un algorithme qui le résout.

Quels sont les problèmes calculables ?

Quelques problèmes calculables

- Additionner deux entiers

Quelques problèmes calculables

- Additionner deux entiers
- Faire un emploi du temps

Quelques problèmes calculables

- Additionner deux entiers
- Faire un emploi du temps
- Compresser/Décompresser un fichier

Quelques problèmes calculables

- Additionner deux entiers
- Faire un emploi du temps
- Compresser/Décompresser un fichier
- Colorier une carte avec le moins de couleurs possibles sans que deux pays voisins aient la même couleur

Quelques problèmes calculables

- Additionner deux entiers
- Faire un emploi du temps
- Compresser/Décompresser un fichier
- Colorier une carte avec le moins de couleurs possibles sans que deux pays voisins aient la même couleur
- ...

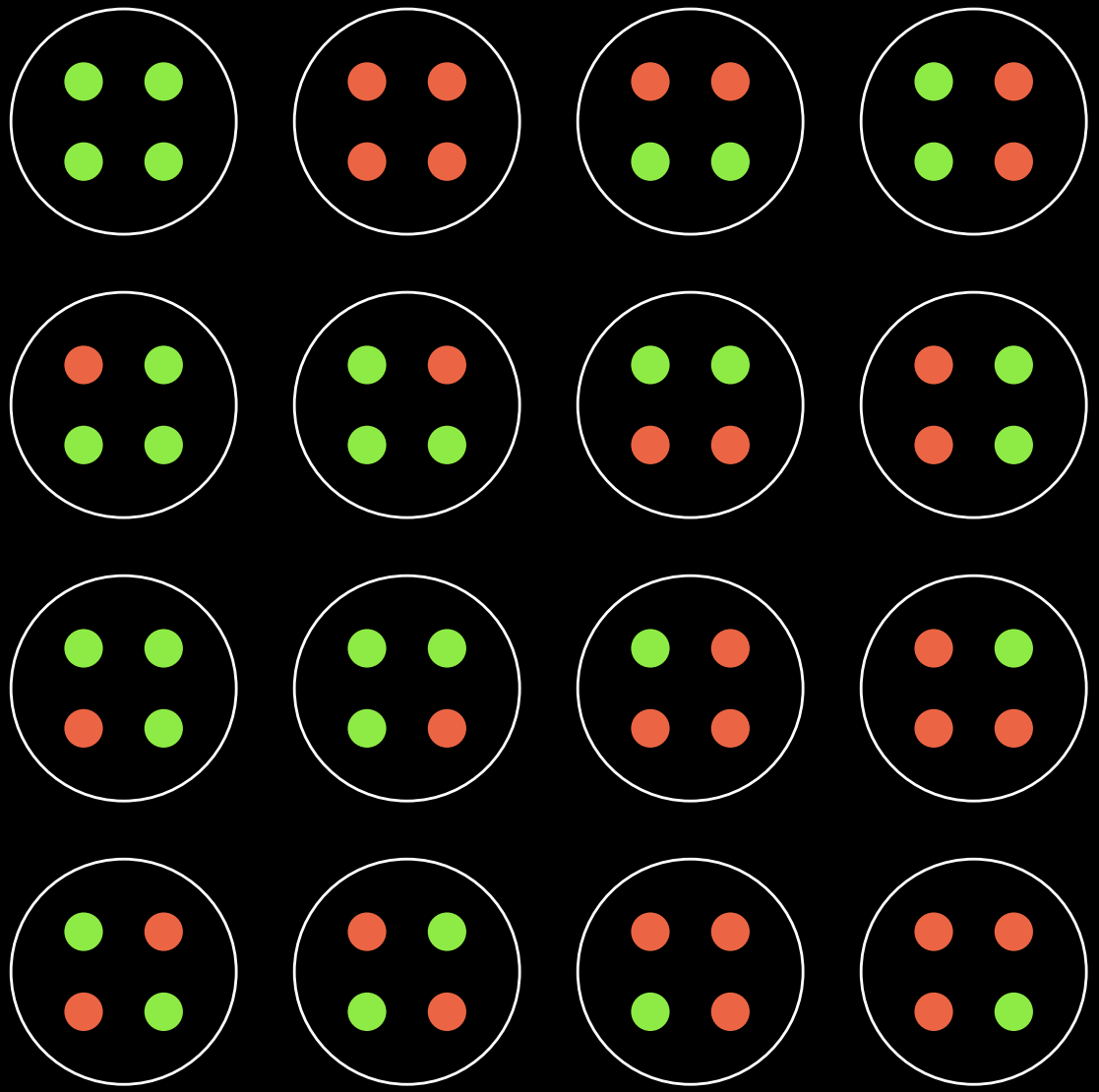
Interlude : le barman aveugle avec des gants de boxe

Le problème

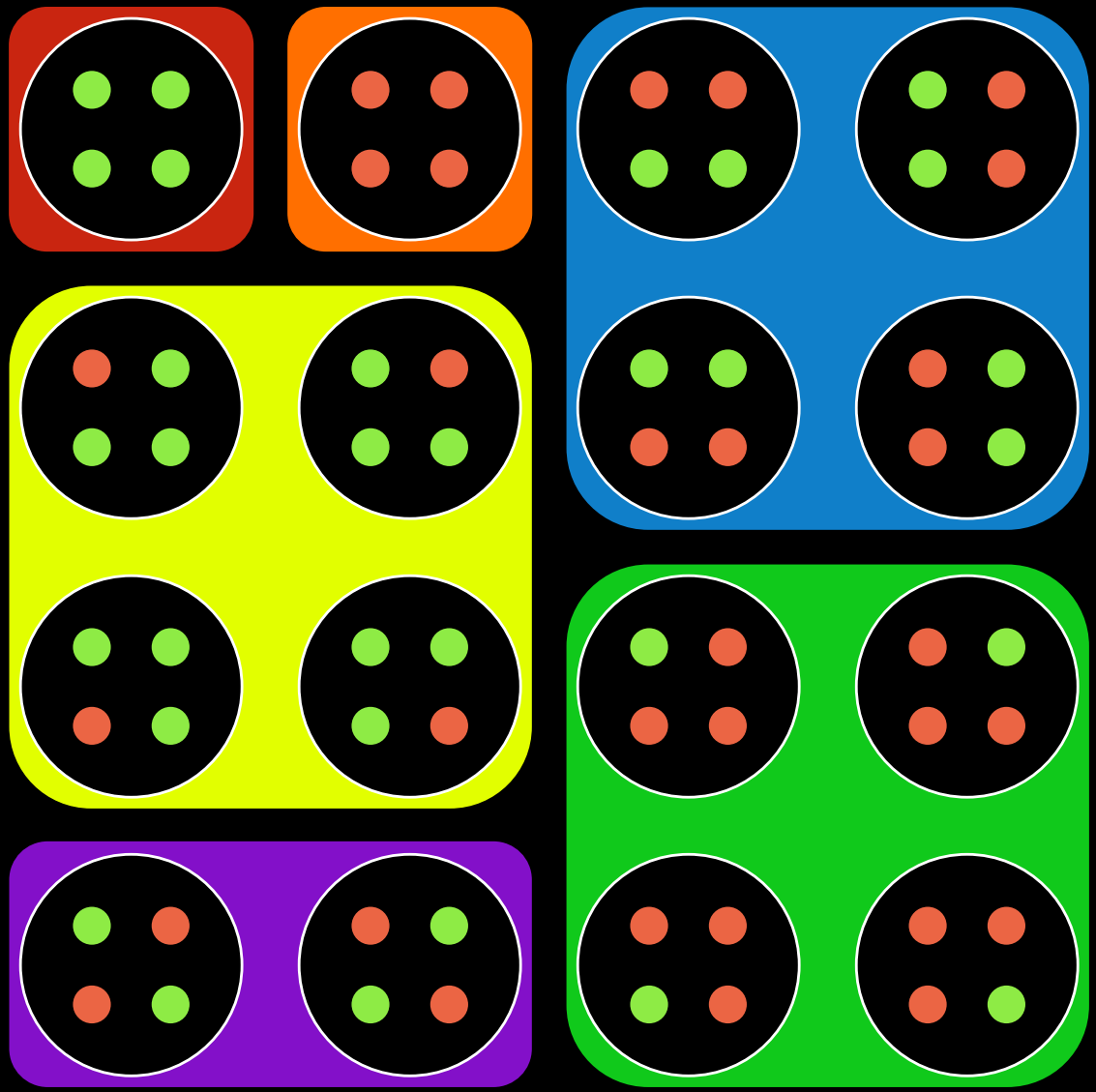
Un barman aveugle a devant lui quatre verres sur un plateau, à l'endroit ou à l'envers. Il doit les mettre tous dans le même sens.
A chaque tour,

- le client tourne le plateau de 0, 90°, 180° ou 270° ;
- le barman retourne des verres
- le client lui dit s'il a réussi

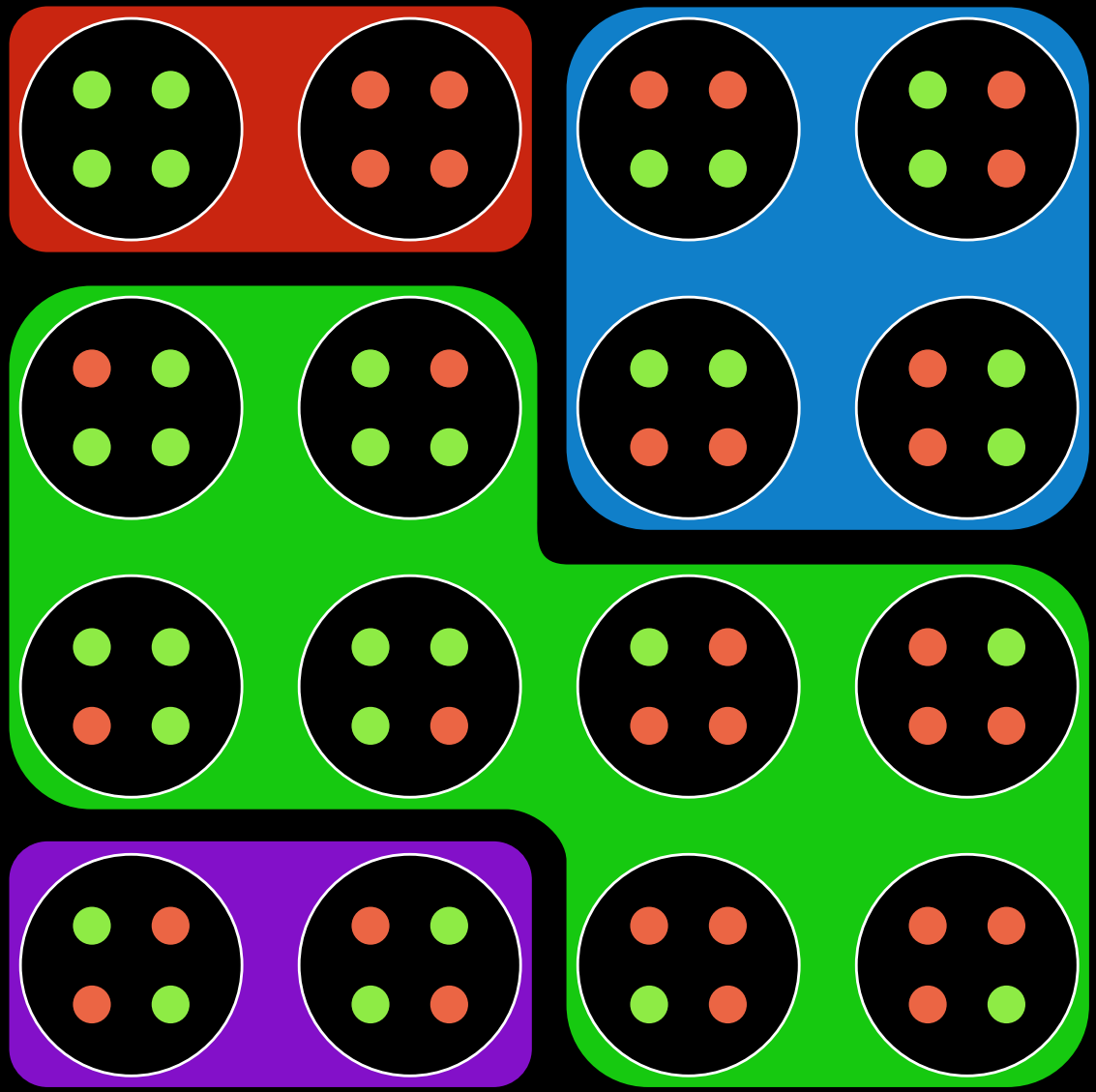
Le barman peut-il réussir ?



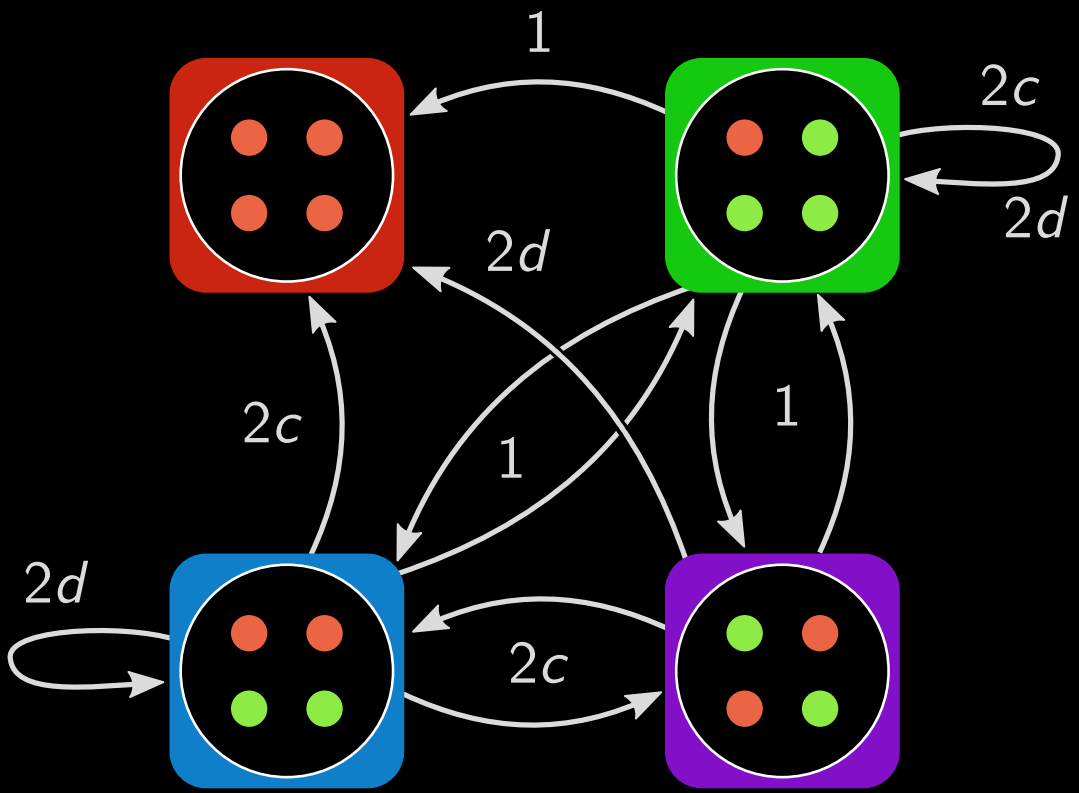
Le barman peut-il réussir ?



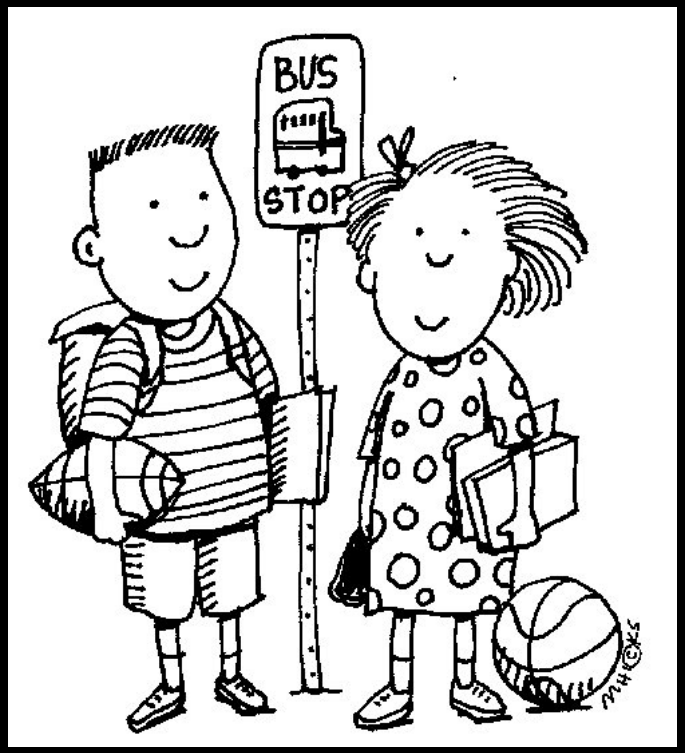
Le barman peut-il réussir ?



Le barman peut-il réussir ?



L'arrêt de bus



On attend le résultat du programme

L'arrêt de bus

Le programme termine

L'arrêt de bus

Le programme continue de calculer

L'arrêt de bus

Le programme continue de calculer

Peut-on savoir si le bus va finir par arriver ?

L'arrêt de programme

PGCD(A, B)

1. $P \leftarrow A$, $Q \leftarrow B$, $R \leftarrow Q$
2. **Tant que** $R > 0$ **Faire**
3. $\quad R \leftarrow$ RESTE(P, Q)
4. $\quad P \leftarrow Q$, $Q \leftarrow R$
5. **Répondre** P

L'arrêt de programme

PGCD(A, B)

1. $P \leftarrow A$, $Q \leftarrow B$, $R \leftarrow Q$
2. **Tant que** $R > 0$ **Faire**
3. $\quad R \leftarrow$ RESTE(P, Q)
4. $\quad P \leftarrow Q$, $Q \leftarrow R$
5. **Répondre** P

COLLATZ(A)

1. **Tant que** $R > 0$ **Faire**
2. $\quad$ **Si** A est pair
3. $\quad$ **Alors** $A \leftarrow A/2$
4. $\quad$ **Sinon** $A \leftarrow 3A + 1$
5. **Répondre** A

L'arrêt de programme

Pgcd(A, B)

1. $P \leftarrow A$, $Q \leftarrow B$, $R \leftarrow Q$
2. **Tant que** $R > 0$ **Faire**
3. $\quad R \leftarrow$ Reste(P, Q)
4. $\quad P \leftarrow Q$, $Q \leftarrow R$
5. **Répondre** P

Collatz(A)

1. **Tant que** $R > 0$ **Faire**
2. $\quad$ **Si** A est pair
3. $\quad$ **Alors** $A \leftarrow A/2$
4. $\quad$ **Sinon** $A \leftarrow 3A + 1$
5. **Répondre** A

On veut un programme Stop tel que

$$\text{Stop}(\mathcal{P}, e) = \begin{cases} \text{Oui} & \text{si } \mathcal{P} \text{ s'arrête sur l'entrée } e, \\ \text{Non} & \text{sinon.} \end{cases}$$

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

Que fait Non-Stop(Non-Stop) ?

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

Que fait Non-Stop(Non-Stop) ?

- Si Stop(Non-Stop, Non-Stop) = Oui :

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

Que fait Non-Stop(Non-Stop) ?

- Si Stop(Non-Stop, Non-Stop) = Oui :
  - Stop nous dit "Non-Stop(Non-Stop) s'arrête".

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

Que fait Non-Stop(Non-Stop) ?

- Si Stop(Non-Stop, Non-Stop) = Oui :
  - Stop nous dit "Non-Stop(Non-Stop) s'arrête".
  - Donc Non-Stop(Non-Stop) ne s'arrête pas !

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

Que fait Non-Stop(Non-Stop) ?

- Si Stop(Non-Stop, Non-Stop) = Oui :
  - Stop nous dit "Non-Stop(Non-Stop) s'arrête".
  - Donc Non-Stop(Non-Stop) ne s'arrête pas !
- Si Stop(Non-Stop, Non-Stop) = Non :

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

Que fait Non-Stop(Non-Stop) ?

- Si Stop(Non-Stop, Non-Stop) = Oui :
  - Stop nous dit "Non-Stop(Non-Stop) s'arrête".
  - Donc Non-Stop(Non-Stop) ne s'arrête pas !
- Si Stop(Non-Stop, Non-Stop) = Non :
  - Stop nous dit "Non-Stop(Non-Stop) ne s'arrête pas".

Il n'y a pas de Stop !

Non-Stop($\mathcal{P}$)

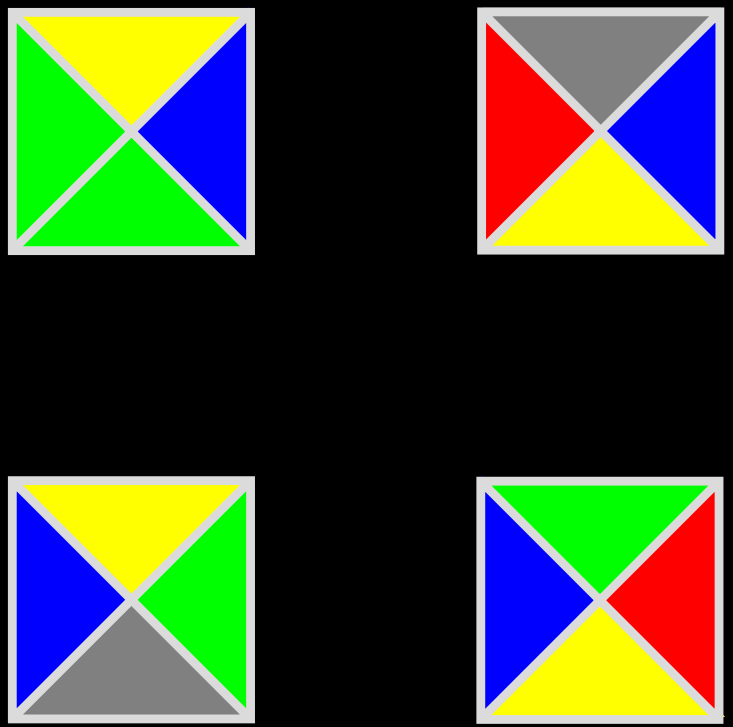
1. **Si** Stop($\mathcal{P}, \mathcal{P}$) = Oui
2. **Alors Boucler infiniment**
3. **Sinon Répondre** Non

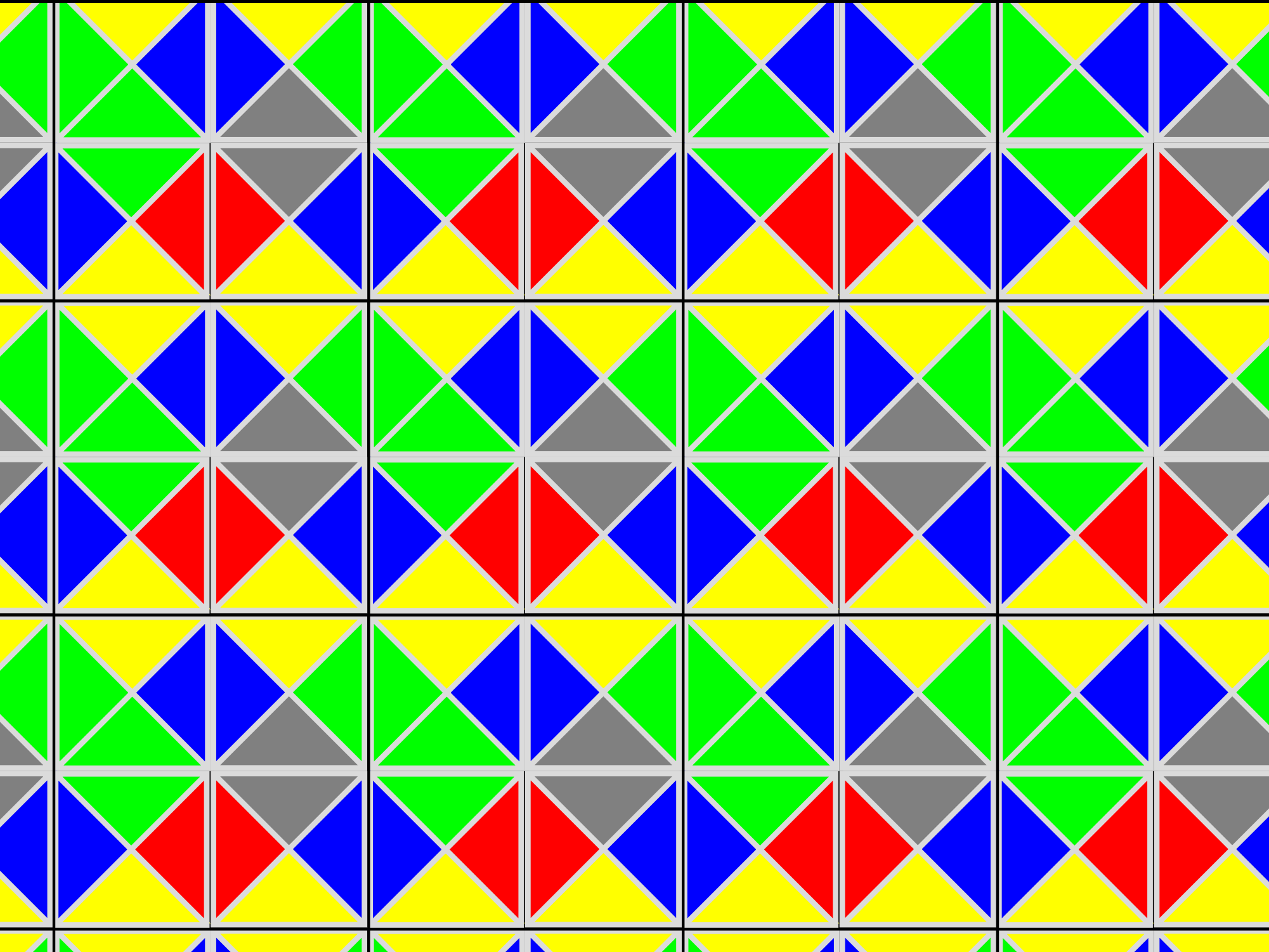
Que fait Non-Stop(Non-Stop) ?

- Si Stop(Non-Stop, Non-Stop) = Oui :
  - Stop nous dit "Non-Stop(Non-Stop) s'arrête".
  - Donc Non-Stop(Non-Stop) ne s'arrête pas !
- Si Stop(Non-Stop, Non-Stop) = Non :
  - Stop nous dit "Non-Stop(Non-Stop) ne s'arrête pas".
  - Donc Non-Stop(Non-Stop) s'arrête !

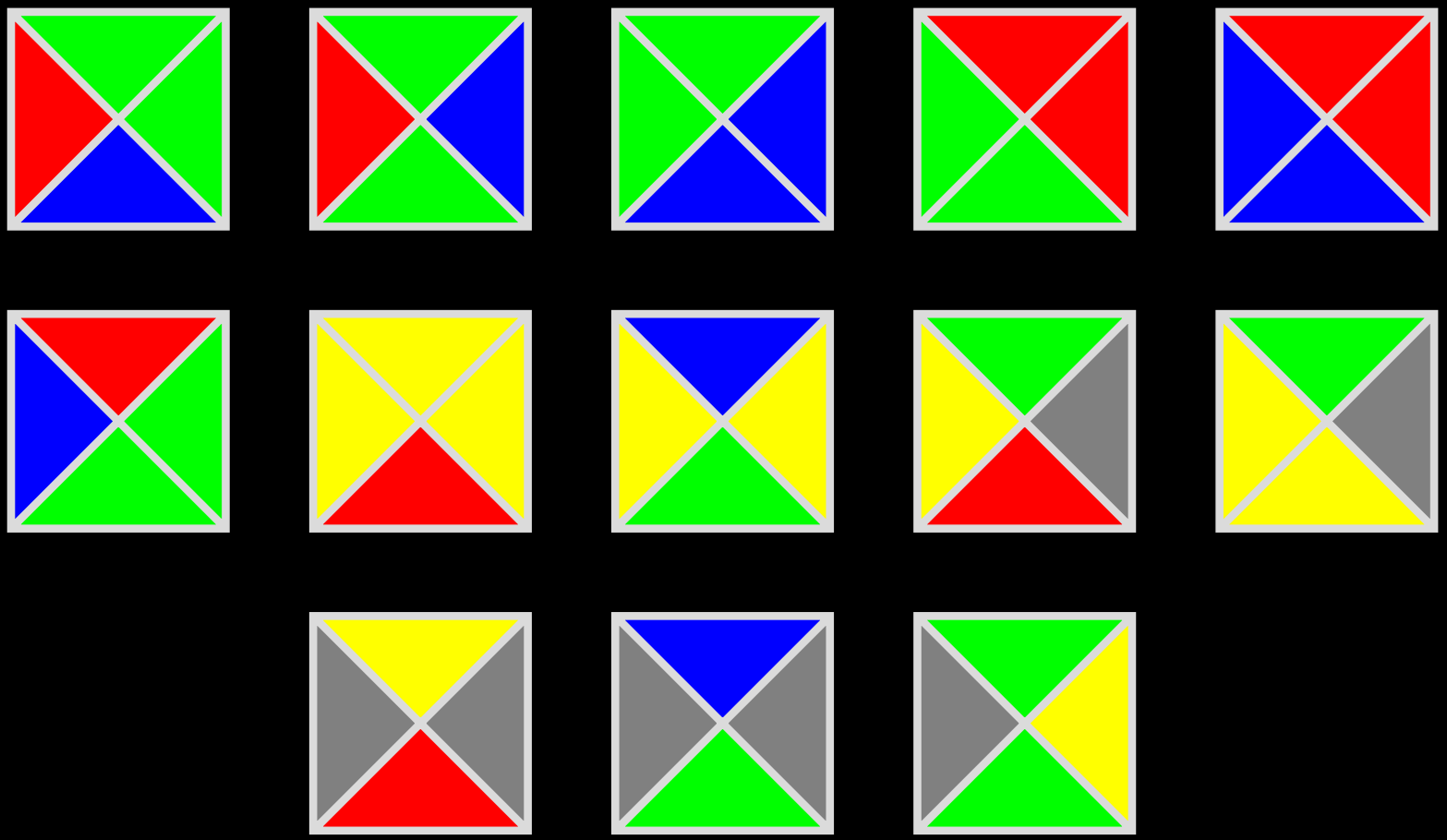
Un problème indécidable : les tuiles de Wang

Peut-on faire un pavage infini avec ces tuiles ?

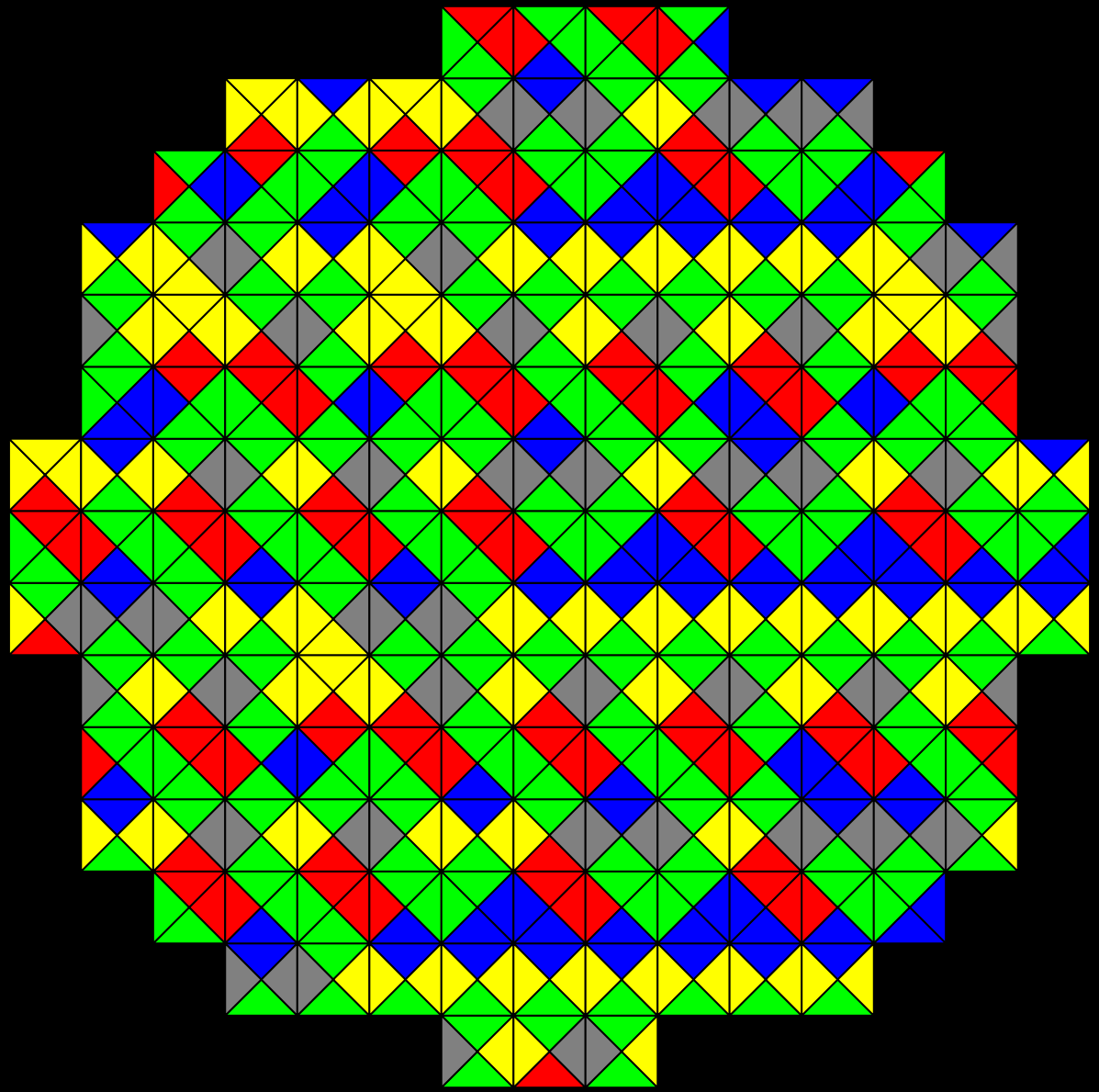




Et avec ces tuiles ?



Et avec ces tuiles ?



La complexité

Question

Trouver p et q tels que

$$1591 = p \times q.$$

La complexité

Question

Trouver p et q tels que

$$1591 = p \times q.$$

Vérifier que

$$1591 = 37 \times 43.$$

La complexité

Question

Trouver p et q tels que

$$1591 = p \times q.$$

Vérifier que

$$1591 = 37 \times 43.$$

Est-ce que trouver est plus difficile que vérifier ?

La complexité

Question

Trouver p et q tels que

$$1591 = p \times q.$$

Vérifier que

$$1591 = 37 \times 43.$$

Est-ce que trouver est plus difficile que vérifier ?

$\rightsquigarrow$ \$1 000 000 !

Formalisation

- Difficulté : nombre d'étapes de calcul (additions, multiplications, etc...)

Formalisation

- Difficulté : nombre d'étapes de calcul (additions, multiplications, etc...)
- Mesure en fonction de la *taille de l'entrée* : nombre de chiffres

Formalisation

- Difficulté : nombre d'étapes de calcul (additions, multiplications, etc...)
- Mesure en fonction de la *taille de l'entrée* : nombre de chiffres

Théorème

Il existe des problèmes faciles à vérifier mais *a priori* durs à résoudre : si on en résout un, on les résout tous !

Stephen A Cook

Formalisation

- Difficulté : nombre d'étapes de calcul (additions, multiplications, etc...)
- Mesure en fonction de la *taille de l'entrée* : nombre de chiffres

Théorème

Il existe des problèmes faciles à vérifier mais *a priori* durs à résoudre : si on en résout un, on les résout tous !

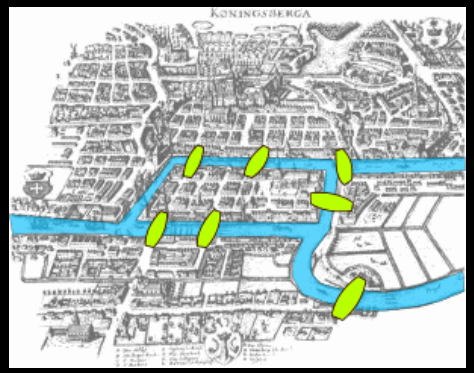
Question à $1 000 000

Sont-ils vraiment difficiles à résoudre ?

Stephen A Cook

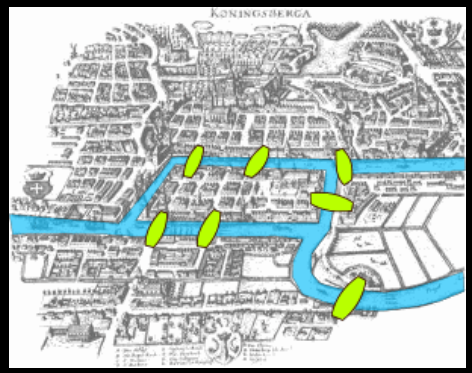
Un problème facile : les ponts de Königsberg

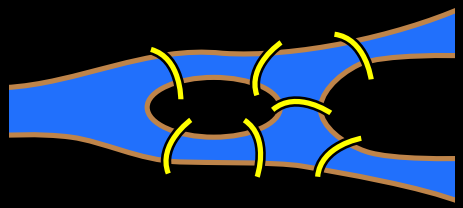
Peut-on passer une fois et une seule par chaque pont ?



Un problème facile : les ponts de Königsberg

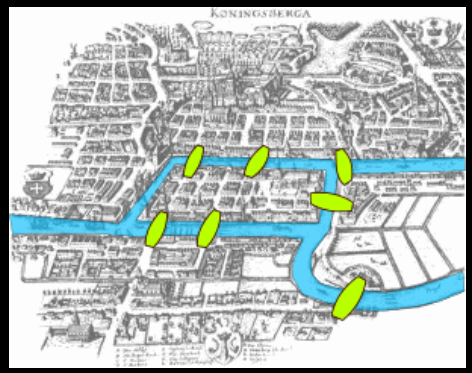
Peut-on passer une fois et une seule par chaque pont ?

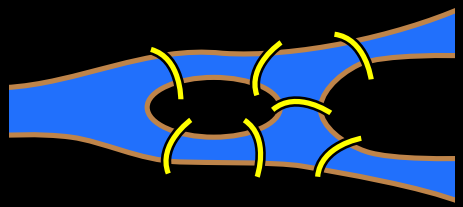


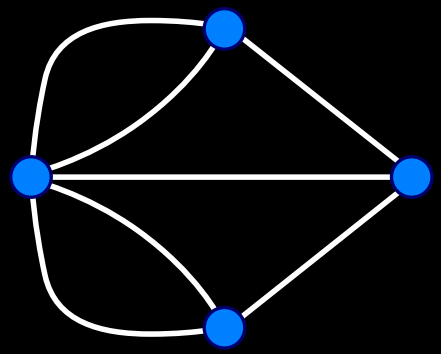


Un problème facile : les ponts de Königsberg

Peut-on passer une fois et une seule par chaque pont ?

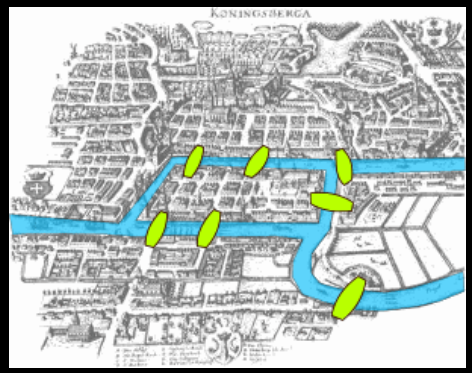


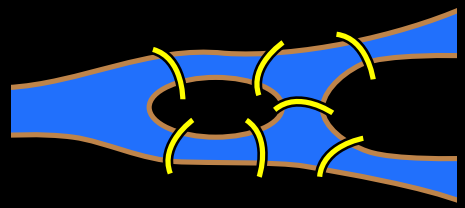


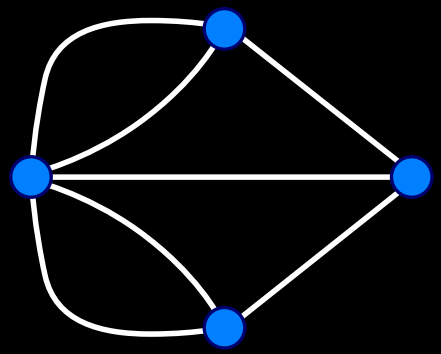


Un problème facile : les ponts de Königsberg

Peut-on passer une fois et une seule par chaque pont ?





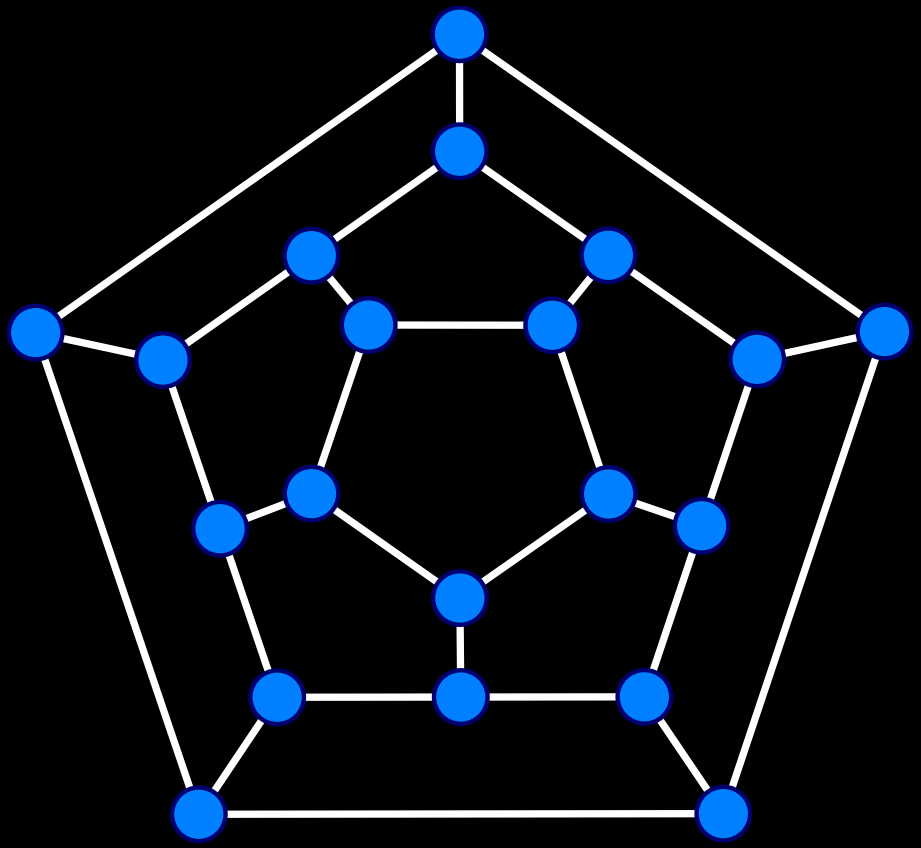


Réponse

C'est possible si et seulement au plus deux sommets ont un degré impair.

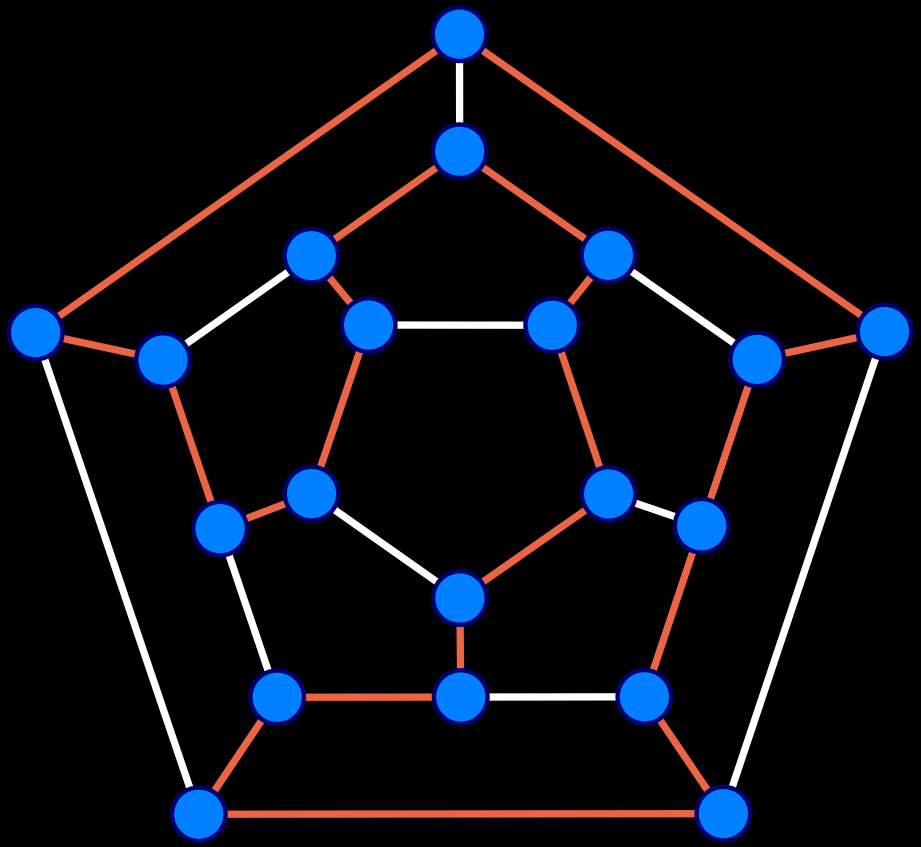
Un problème difficile : le chemin hamiltonien

Peut-on passer une fois et une seule chaque sommet ?



Un problème difficile : le chemin hamiltonien

Peut-on passer une fois et une seule chaque sommet ?



La Cryptographie

$$\text{Message } m \xrightarrow{\text{codage}} \text{Message codé } c(m) \xrightarrow{\text{décodage}} \text{Message } m$$

La Cryptographie

Message m $\xrightarrow{\text{codage}}$ Message codé $c(m)$ $\xrightarrow{\text{décodage}}$ Message m

- Codage simple :

m	a	b	c	$\cdots$	k	$\cdots$
$c(m)$	q	r	s	$\cdots$	a	$\cdots$

La Cryptographie

Message m $\xrightarrow{\text{codage}}$ Message codé $c(m)$ $\xrightarrow{\text{décodage}}$ Message m

- Codage simple :

m	a	b	c	$\cdots$	k	$\cdots$
$c(m)$	q	r	s	$\cdots$	a	$\cdots$

- Codage avec clef :

Exemple : le codage de Vigénère

m	M	e	s	s	a	g	e		s	e	c	r	e	t
Clef	m	a	g	i	k	m	a		g	i	k	m	a	g
	13	1	7	9	11	13	1		7	9	11	13	1	7
$c(m)$	Z	f	z	b	l	t	f		z	n	n	e	f	a

La cryptographie est-elle possible ?

Théorème

Il existe des fonctions de codage faciles à calculer mais *a priori* dures à décoder.

La cryptographie est-elle possible ?

Théorème

Il existe des fonctions de codage faciles à calculer mais *a priori* dures à décoder.

Ces fonctions de codage sont vraiment dures à décoder si et seulement s'il y a des problèmes faciles à vérifier qui sont vraiment dur à résoudre.

Les limites de la compression de données

Les limites de la compression de données

Fichier f $\xrightarrow{\text{compression}}$ Fichier compressé $f.zip$ $\xrightarrow{\text{décompression}}$ Fichier f

- Compression *sans perte* $\neq$ compression d'images, photos, etc...

Théorème

Il y a des fichiers incompressibles !

Andrey Nikolaevich Kolmogorov

Des questions ?