Bonus

Pascal Lafourcade

UNIVERSITÉ Clermont Auvergne

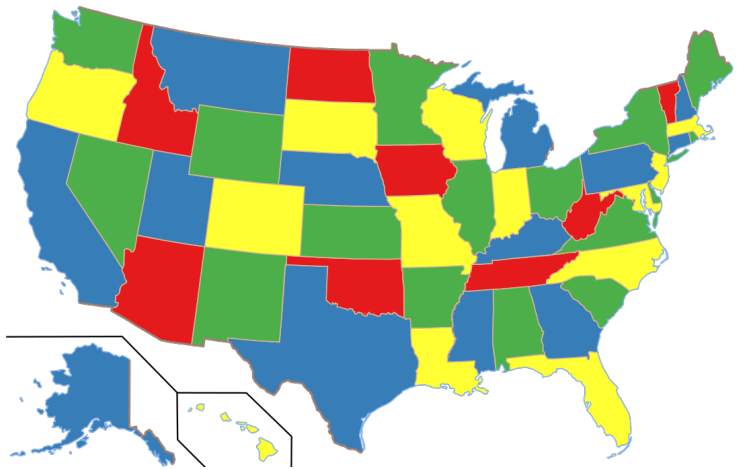
LIMOS

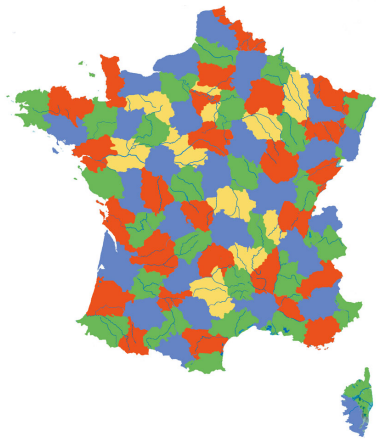
R5.A04 Qualité algorithmique
BUT 3, 2023-2024

Coloriage de cartes : Théorème de 4 couleurs

Conjecture en 1840 par Ferdinand Möbius

Il est toujours possible de colorier une carte connexe avec 4 couleurs tel que deux régions adjacentes sont de couleurs distinctes.

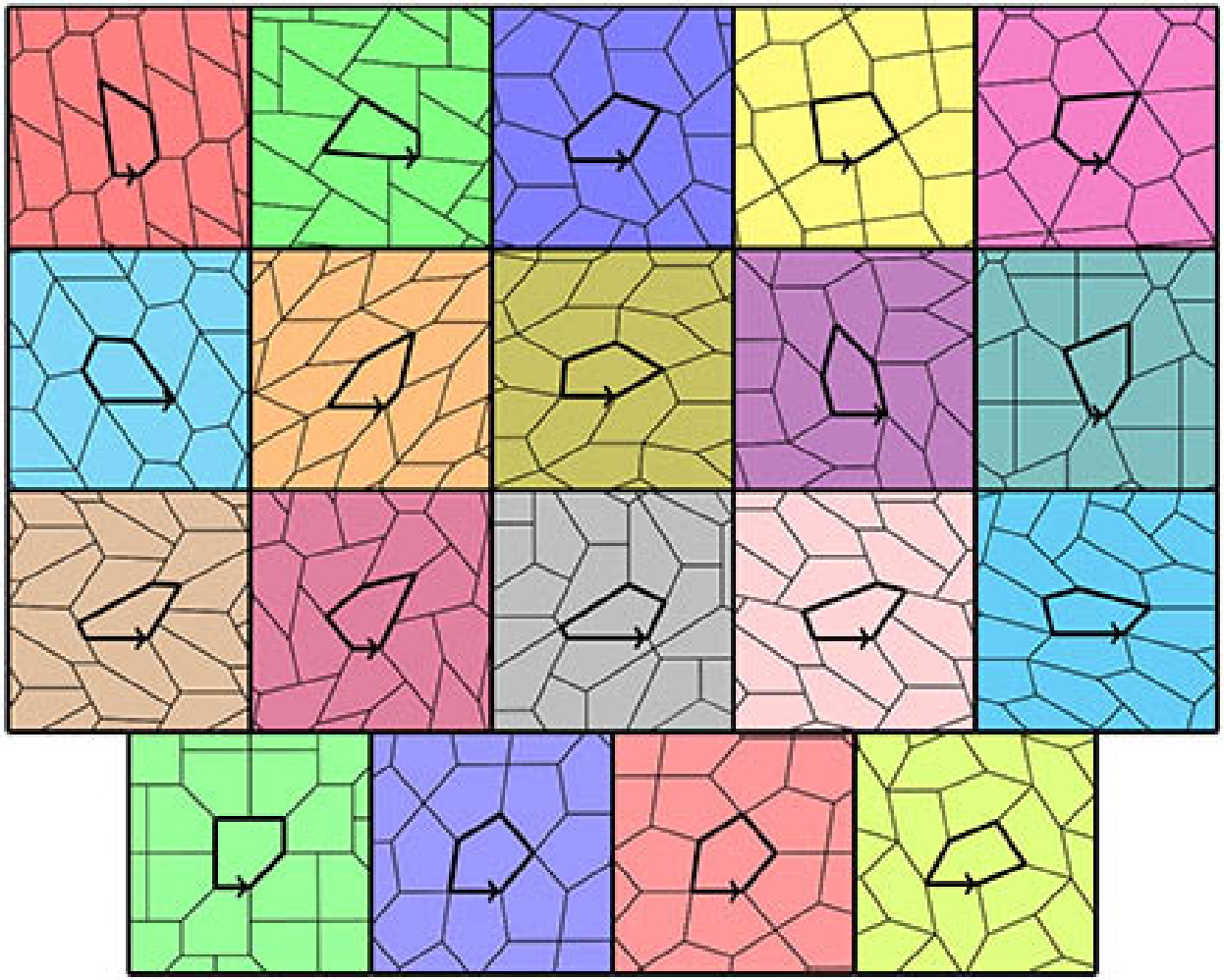




En 1879, preuve pour 5 couleurs par Alfred Kempe.
En 1976, preuve avec un ordinateur (1 478 cas critiques) par Kenneth Appel et Wolfgang Haken.
En 2005, preuve par Georges Gonthier et Benjamin Werner.

2017 Pavage du plan : 15 possibilités

Michael Rao, ENS LYON



Seul 15 famille de polynomes convexes pavent le plan.

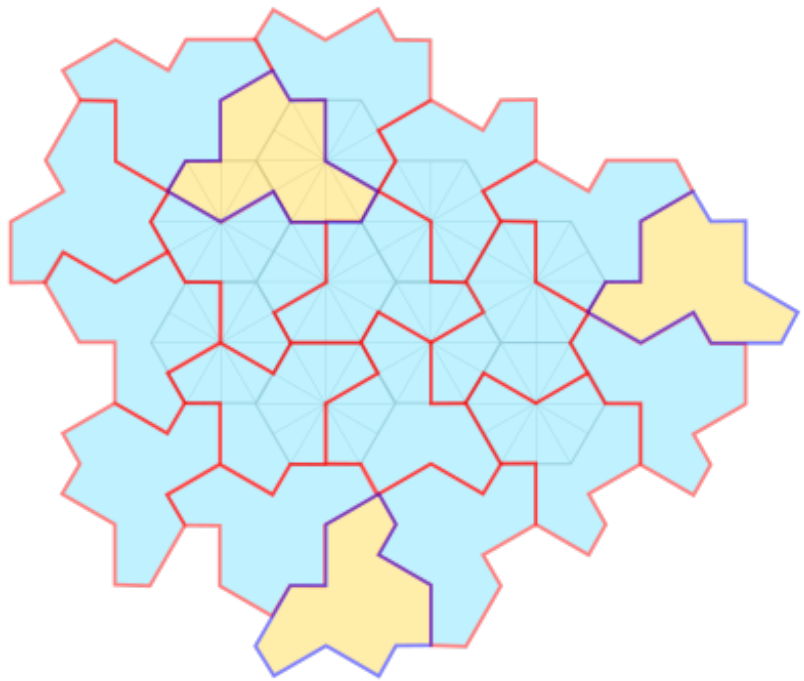
`https://arxiv.org/abs/1708.00274`

2023: An aperiodic monotile

David Smith, Joseph Samuel Myers, Craig S. Kaplan, Chaim Goodman-Strauss

Einstein

A shape that admits tilings of the plane, but never periodic tilings.

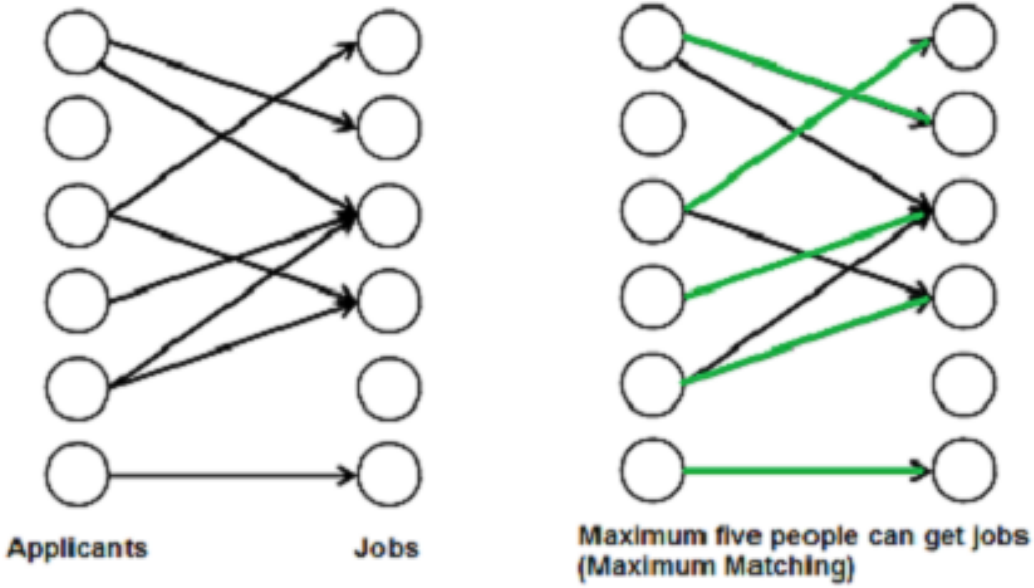


Computer-assisted proof

Maximum Bipartite Matching

Problem

A matching in a Bipartite Graph is a set of the edges chosen in such a way that no two edges share an endpoint. A maximum matching is a matching of maximum size (maximum number of edges). In a maximum matching, if any edge is added to it, it is no longer a matching. There can be more than one maximum matchings for a given Bipartite Graph.



Maximum Bipartite Matching

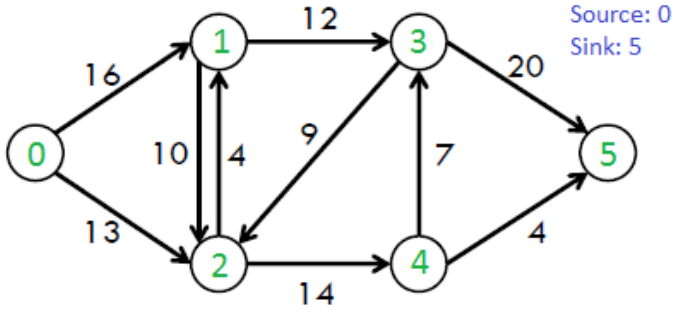
Year	Authors	Time Bound & Notes
	folklore/trivial	mn BIPARTITE
1965 1965	Edmonds Witzgall & Zahn	mn^2
1969 1976 1976 1976	Balinski Gabow Lawler Karzanov	mn or n^3
1971 1973	Hopcroft & Karp Dinic & Karzanov	$m\sqrt{n}$ BIPARTITE
1980 1991	Micali & Vazirani Gabow & Tarjan	$m\sqrt{n}$
1981	Ibarra & Moran	n^ω CARDINALITY ONLY,RANDOMIZED,BIPARTITE
1989	Rabin & Vazirani	n^ω CARDINALITY ONLY,RANDOMIZED $n^{\omega+1}$ RANDOMIZED
1991	Alt, Blum, Mehlhorn & Paul	$n\sqrt{nm/\log n}$ BIPARTITE
1991 1997	Feder & Motwani Goldberg & Kennedy	$m\sqrt{n}/\kappa$ BIPARTITE, $\kappa = \frac{\log n}{\log(n^2/m)}$
1996	Cheriyan & Mehlhorn	$n^2 + n^{5/2}/w$ BIPARTITE, w = machine word size
2004	Goldberg & Karzanov	$m\sqrt{n}/\kappa$
2004 2006	Mucha & Sankowski Harvey	n^ω RANDOMIZED

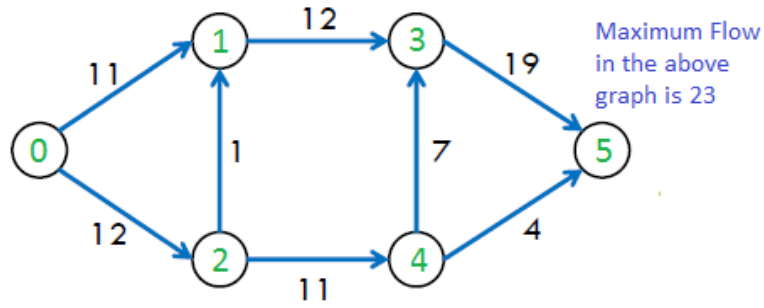
Note: Here $\omega < 2.373$ is the exponent of $n \times n$ matrix multiplication [Williams 2012]. The mn running time on general graphs depends on a special union-find data structure [Gabow and Tarjan 1985] developed later. Without it, the running time would be $mn\alpha(m,n)$, where α is the inverse-Ackermann function.

En 2024 : Maximum Bipartite Matching $n^{2+o(1)}$

A Faster Combinatorial Algorithm for Maximum Bipartite Matching.
Julia Chuzhoy, Sanjeev Khanna (SODA)

Maximum Flow & Minimum Cost-Flow 2022





Maximum FLow

Un graphe orienté $G = (V, E)$ où chaque arête de E possède une capacité et peut recevoir un flot (ou flux). Le cumul des flots sur une arête ne peut pas excéder sa capacité. Calculer un flot réalisable depuis une source unique et vers un puits unique qui soit maximum.

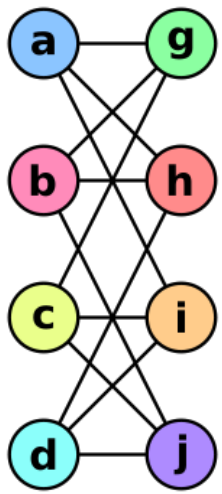
- 1970 : Algorithme de flot bloquant de Dinitz $O(V^2E)$
- 1972 : Algorithme d'Edmonds-Karp $O(VE^2)$
- 1994 : Algorithme de King, Rao et Tarjan $O(EV \log_{\frac{E}{V \log V}} V)$
- 2013 : Algorithme de James B Orlin $O(VE)$
- 2022 : Maximum Flow and Minimum-Cost Flow in Almost-Linear Time $O(E^{1+o(1)})$, Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg,

Isomorphisme de Graphe

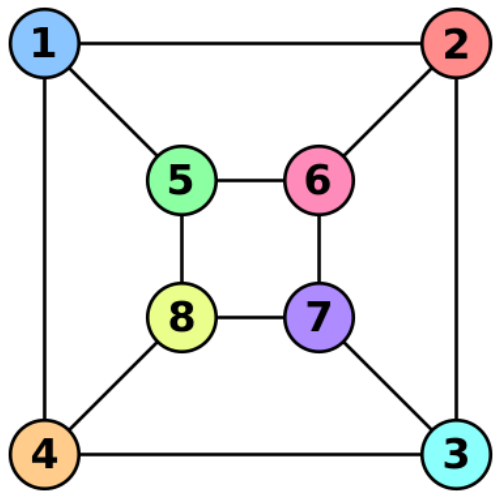
Definition

Un isomorphisme de graphes est une bijection entre les sommets de deux graphes qui préserve les arêtes.

Graph G



Graph H



Isomorphisme entre G et H

f(a)=1
f(b)=6
f(c)=8
f(d)=3
f(g)=5
f(h)=2
f(i)=4
f(j)=7

Question ouverte (2024)

Étant donné deux graphes à savoir s'ils sont isomorphes, est dans NP mais on ne connait ni d'algorithme en temps polynomial, ni de preuve qu'il est NP-complet.

Isomorphisme de Graphe

- Babai, Kantor and Luks (FOCS 1983) $exp(O(\sqrt{n}logn))$
- LászlBabai (STOC 2016) quasipolynomial($exp((logn)O(1))$)
- Martin Grohe; Daniel Neuen; Pascal Schweitzer (FOCS 2018) *A Faster Isomorphism Test for Graphs of Small Degree* $n^{O((logd)^c}$

Exponetial Time Hypothesis (Impagliazzo & Paturi (CCC 1999))

Satisfiability of 3-CNF Boolean formulas cannot be solved in subexponential time, $2^{o(n)}$.

Cela implique P $\neq$ NP.
Donc l'idée est que l'isomorphisme de graphe ne soit pas NP.

Algorithme de Strassen 1986

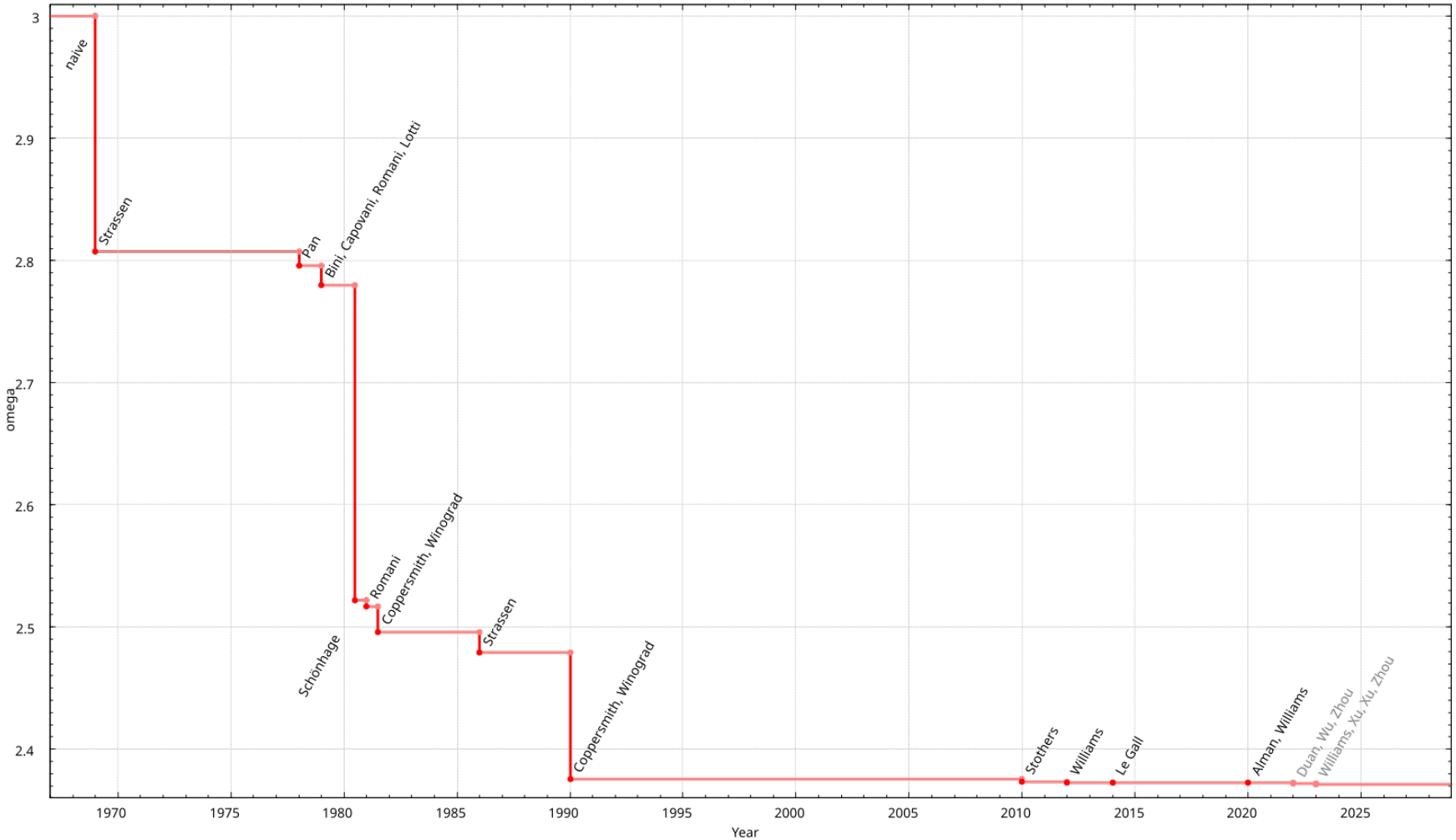
$A, B \in \mathbb{R}^{d \times d}$ où $d := 2^k$ et $k \in \mathbb{N}^*$.

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}, B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}, C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$$

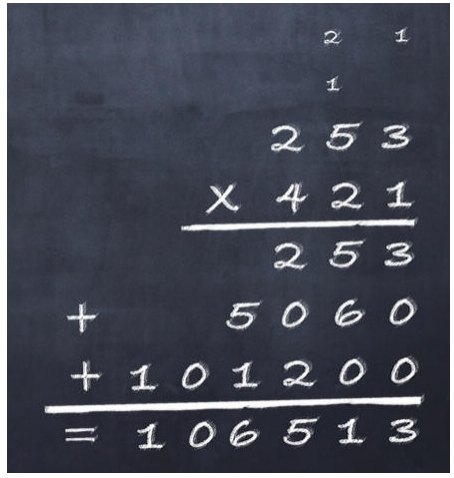
$M_1 := (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2})$
$M_2 := (A_{2,1} + A_{2,2})B_{1,1}$
$M_3 := A_{1,1}(B_{1,2} - B_{2,2})$
$M_4 := A_{2,2}(B_{2,1} - B_{1,1})$
$M_5 := (A_{1,1} + A_{1,2})B_{2,2}$
$M_6 := (A_{2,1} - A_{1,1})(B_{1,1} + B_{1,2})$
$M_7 := (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2})$
$C_{1,1} = M_1 + M_4 - M_5 + M_7$
$C_{1,2} = M_3 + M_5$
$C_{2,1} = M_2 + M_4$
$C_{2,2} = M_1 - M_2 + M_3 + M_6$

7 multiplications 18 additions !

Multiplication de matrices



Algorithme de multiplication de 2 nombres entiers



$$(a \times 10^k + b)(c \times 10^k + d) = ac \times 10^{2k} + (ad + bc) \times 10^k + bd$$

- Nécessite 4 produits ac, ad, bc et bd.
- Soit $K(n)$ le nombre de multiplications pour le produit de 2 nombres à n chiffres :

$$K(n) < 4K(n/2)$$

- Complexité $O(n^{log_2(4)}) \approx O(n^2)$

Karatsuba, 1960

$$(a\times 10^k+b)(c\times 10^k+d) = ac\times 10^{2k}+(ac+bd-(a-b)(c-d))\times 10^k+bd$$

- 3 produits ac, bd et $(a–b)(c–d)$
- Soit $K(n)$ le nombre de multiplications pour le produit de 2 nombres à n chiffres :

$$K(n) \leq 3K(\lceil n/2 \rceil)$$

- Complexité $O(n^{log_2(3)}) \approx O(n^{1,585})$ au lieu de $O(n^2)$

Autres techniques

- 1960 : Karatsuba, $O(n^{log_2(3)}) \approx O(n^{1,585})$
- 1963 : Algorithme Toom-Cook, $\mathcal{O}(n^{\log_3(5)}) \approx \mathcal{O}(n^{1,465})$
- 1971 : Algorithme de Schönhage-Strassen, $\mathcal{O}(n \cdot \log n \cdot \log\log n)$
- 2007 : Algorithme de Fürer : $\mathcal{O}(n \log n K^{O(\log^* n)})$, où $\log^* n$ désigne le logarithme itéré et K un nombre strictement supérieur à 1
- 2021 : David Harvey, Joris van Der Hoeven : Integer multiplication in time $O(nlogn)$, Annals of Mathematics.

`https://hal.science/hal-02070778/document`

10eme problème de Hilbert

23 problèmes posés par David Hilbert en 1900.

10eme problème de Hilbert

Trouver un algorithme pour résoudre un système d'équations diophantiennes.

En 1970, Youri Matiiassevitch prouve que c'est impossible!

Théorème DPRM (Martin Davis, Hilary Putnam, Julia Robinson et Matiiassevitch)

Un ensemble est récursivement énumérable si et seulement s'il est diophantien.

Un sous-ensemble S de $\mathbb{N}^n$ est dit récursivement énumérable s'il existe un algorithme qui s'exécute indéfiniment et qui énumère exactement tous les membres de S.

Plus court chemin 1A

L'algorithme de Bellman-Ford (1956) est plus général que celui de Dijkstra (1959), mais il est moins rapide.
L'algorithme de Floyd-Warshall (1962) permet de calculer de façon matricielle les plus cours chemins entre tous les couples de sommets d'un graphe pondéré.

Plus court chemin : Dijkstra

Principe

Parcours en largeur à partir du sommet de départ.
Les voisins du sommet initial sont alors visités.
En rangeant les sommets dans une file d'attente, jusqu'à ce que le sommet de destination soit atteint.

Problème du voyageur de commerce

William Rowan Hamilton and Thomas Kirkman

Travelling Salesman Problem (TSP)

Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city.

Algorihtm by brute force is in $O(n!)$.
TSP is NP-Hard thanks to Richard M. Karp.
In 1972, he proves that the Hamiltonian cycle problem was NP-complete.

Problème du voyageur de commerce

The TSP can be formulated as an integer linear program (ILP) $c_{ij} > 0$ to be the cost (distance) from city i to city j.

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise.} \end{cases}$$

Minimiser :

$$\sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}.$$

- Miller–Tucker–Zemlin (MTZ) formulation
- Dantzig–Fulkerson–Johnson (DFJ) formulation

Qu'est-ce ?

Lemme

Si $pgcd(a, n) = 1$, $n > 2$ alors n est premier si et seulement si

$$(X + a)^n = X^n + a \bmod n$$

Preuve:

Si n est premier, alors tous les coeffificents binomiaux sont des multiples de n.

Si n est pas premier, alors q^k divise n c-a-d $n = m \times q^k$.

q^k ne divise pas $\binom{n}{q} = \frac{n \times (n-1) \times \cdots \times (n-q+1)}{q \times (q-1) \times \cdots \times 1}$

q^k est coprime avec a^{n-q}

Donc le coefficient de X^q est non nul.

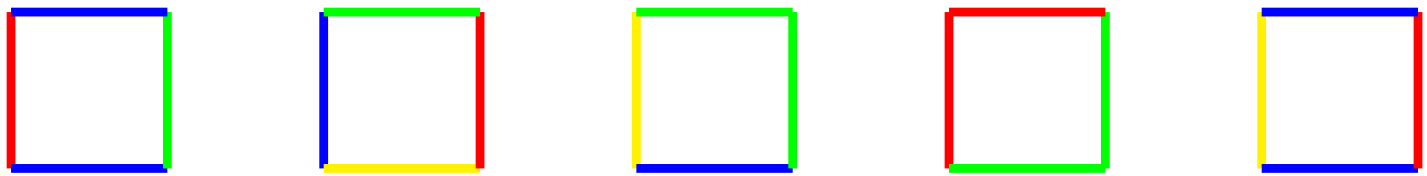
AKS Algorithm

Input: integer $n > 1$.

1. Check if n is a perfect power: if $n = a^b$ for integers $a > 1$ and $b > 1$, then output composite.
2. Find the smallest r such that $ord_r(n) > (log_2 n)^2$. If r and n are not coprime, then output composite.
3. For all $2 \geq a \geq min(r, n-1)$, check that a does not divide n: If $a|n$ for some $2 \geq a \geq min(r, n-1)$, then output composite.
4. If $n \geq r$, then output prime.
5. For $a = 1$ to $\left\lfloor \sqrt{\varphi(r)} \log_2(n) \right\rfloor$ do
   if $(X + a)^n \neq X^n + a (mod X^r - 1, n)$, then output composite;
6. Output prime.

Pavage du plan

- **Instance:** Soit un ensemble fini de carrés colorés.

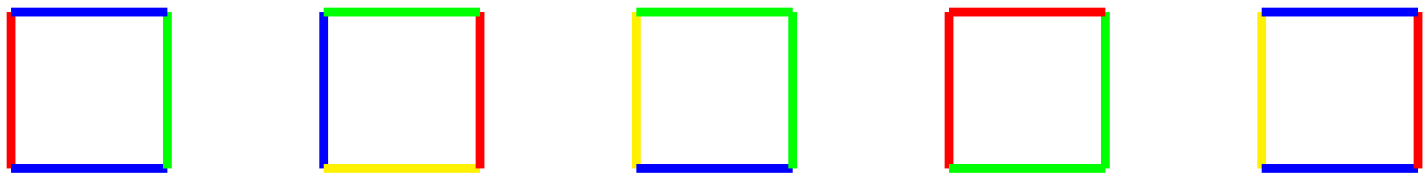


- **Problème:** Décider quel sous ensemble fini peut être utilisé pour paver le plan ?

- **Conditions:** Les arrêtes adjacentes doivent avoir la mêmes couleur (seulemnt avec des translation)

Pavage du plan

- **Instance:** Soit un ensemble fini de carrés colorés.



- **Problème:** Décider quel sous ensemble fini peut être utilisé pour paver le plan ?

- **Conditions:** Les arrêtes adjacentes doivent avoir la mêmes couleur (seulemnt avec des translation)

C'est un problème indécidable !

Perfect Antivirus cannot exist

Virus Detection is Undecidable

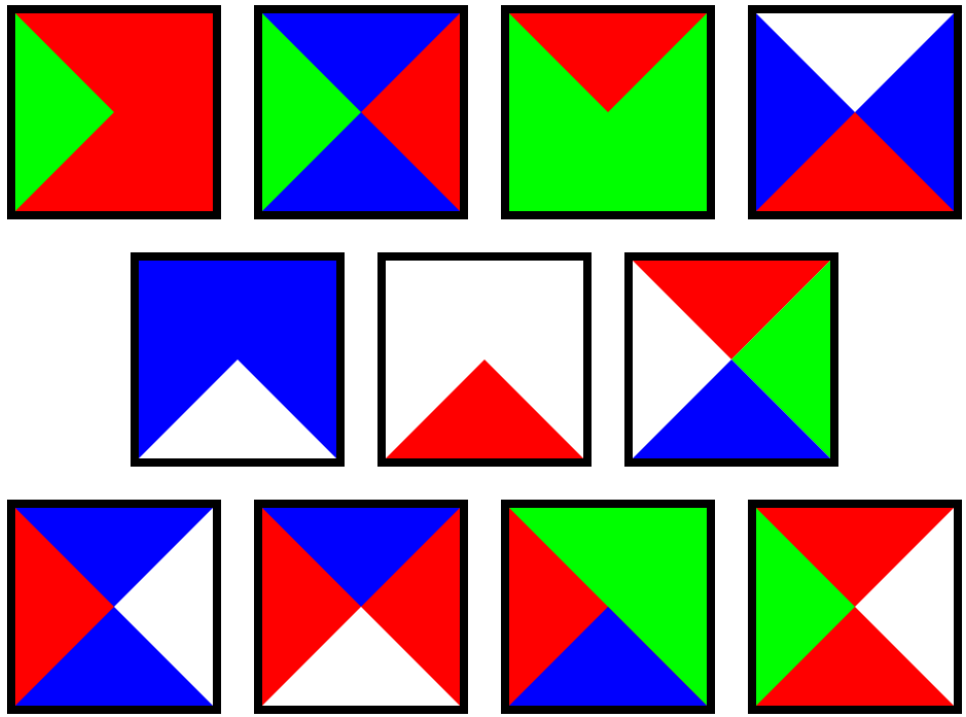
Theorem by Fred Cohen (1987)

Virus abstractly modeled as program that eventually executes infect Code where infect may be generated at runtime

Proof by contradiction similar to that of the halting problem.
Suppose isVirus (P) determines whether program P is a virus
Define new program Q as follows:
Q: if (not isVirus (Q)) then Q infects else Q stops
Running isVirus on Q achieves a contradiction, two cases

- isVirus(Q) is true $\Rightarrow$ Q does nothing
- isVirus(Q) is false $\Rightarrow$ Q infects

Pavage de Hao Wang, 1966



Problème Indécidable

Un ensemble donnée de tuiles de Wang peut-il paver le plan ?

Théorèmes de Goëdel, 1931

Théorème d'incomplètude

Dans n'importe quelle théorie récursivement axiomatisable, cohérente et capable de " formaliser l'arithmétique ", on peut construire un énoncé arithmétique qui ne peut être ni prouvé ni réfuté dans cette théorie.

Théorème d'incomplètude

Si T est une théorie cohérente qui satisfait des hypothèses analogues, la cohérence de T, qui peut s'exprimer dans la théorie T, n'est pas démontrable dans T.

Preuve simplifiée

Machine de Turing

Pour tout programme de taille finie, la machine de Truing répond VRAI ou FAUX à une affirmation qu'on lui donne, sans jamais se tromper.

Que signifie alors le théorème de Goëdel

Que signifie alors le théorème de Goëdel

Si un humain est capable de savoir si la phrase qu'il donne à la machine est vraie ou fausse, la machine est-elle aussi capable de découvrir la vérité ?

La phrase de Goëdel

"La machine ne répondra jamais VRAI à cette phrase"

La preuve ou que fait la machine ?

"La machine ne répondra jamais VRAI à cette phrase"

Deux réponses sont possibles

- VRAI
- FAUX

La preuve ou que fait la machine ?

"La machine ne répondra jamais VRAI à cette phrase"

VRAI

"La machine ne répondra jamais VRAI à cette phrase" est donc une affirmation vraie.

La preuve ou que fait la machine ?

"La machine ne répondra jamais VRAI à cette phrase"

VRAI

"La machine ne répondra jamais VRAI à cette phrase" est donc une affirmation vraie.
Si la machine ne se trompe pas, elle ne peut donc pas répondre VRAI.

La preuve ou que fait la machine ?

"La machine ne répondra jamais VRAI à cette phrase"

VRAI

"La machine ne répondra jamais VRAI à cette phrase" est donc une affirmation vraie.
Si la machine ne se trompe pas, elle ne peut donc pas répondre VRAI.

FAUX

"La machine ne répondra jamais VRAI à cette phrase" est une affirmation fausse.

La preuve ou que fait la machine ?

"La machine ne répondra jamais VRAI à cette phrase"

VRAI

"La machine ne répondra jamais VRAI à cette phrase" est donc une affirmation vraie.
Si la machine ne se trompe pas, elle ne peut donc pas répondre VRAI.

FAUX

"La machine ne répondra jamais VRAI à cette phrase" est une affirmation fausse.
Si la machine ne se trompe pas, elle ne peut donc pas répondre FAUX.

La preuve ou que fait un homme?

"La machine ne répondra jamais VRAI à cette phrase"

Nous venons de voir que la machine ne peut pas répondre VRAI.

La preuve ou que fait un homme?

"La machine ne répondra jamais VRAI à cette phrase"

Nous venons de voir que la machine ne peut pas répondre VRAI.
Nous savons aussi que cette phrase est une vérité.

La preuve ou que fait un homme?

"La machine ne répondra jamais VRAI à cette phrase"

Nous venons de voir que la machine ne peut pas répondre VRAI.
Nous savons aussi que cette phrase est une vérité.
Pourtant la machine ne pourra pas la découvrir...

Question dans Don Quichotte:

À la frontière d'un pays, il faut dire la vérité sinon c'est la pendaison

Un garde frontière vous demande:
“ Pourquoi venez-vous ?”
Vous répondez :

Question dans Don Quichotte:

À la frontière d'un pays, il faut dire la vérité sinon c'est la pendaison

Un garde frontière vous demande:
“ Pourquoi venez-vous ?”
Vous répondez :

“ Pour être pendu !”

Don quichotte de Miguel de Cervantes

"Par une ancienne loi de cette île, tout homme qui vient après la retraite sonnée pour passer ce pont est obligé de nous déclarer, sous la foi du serment, où il va. S'il dit la vérité nous le laissons passer sans obstacle ; s'il fait le moindre mensonge, il est pendu sur-le-champ à une potence dressée à l'autre bout de de ce pont. Cette loi est connue de tous les habitants de votre île. Tout à l'heure l'homme que voici s'est presenté pour passer : nous l'avons interrogé suivant l'usage; il a levé la main et nous a répondu qu'il allait se faire pendre à cette potence."

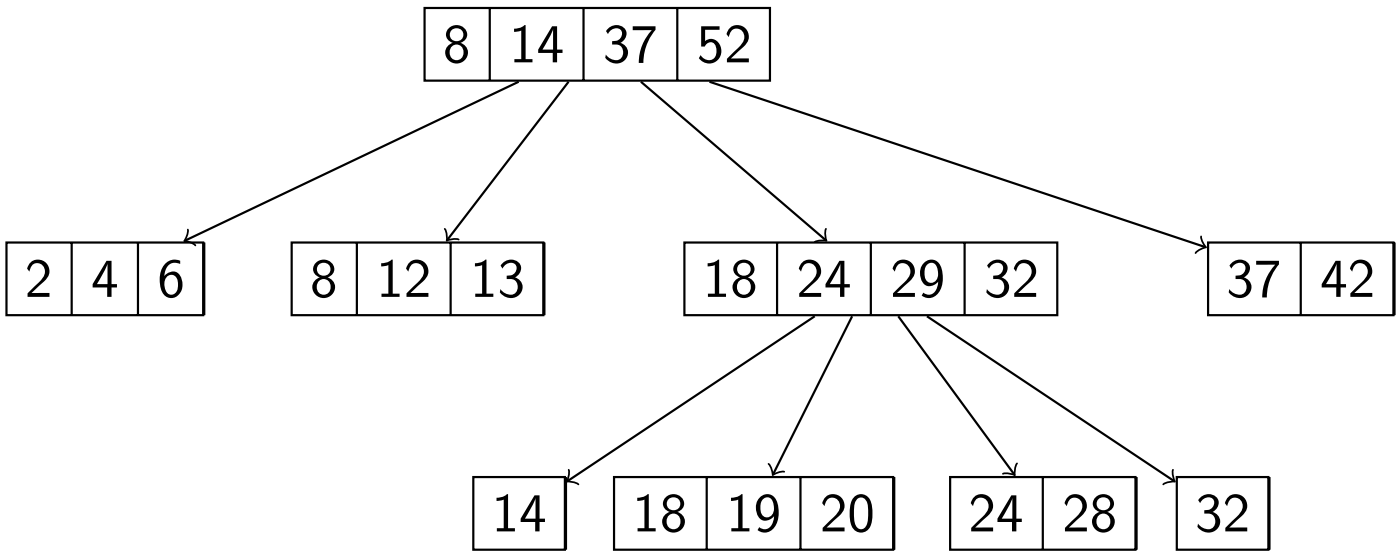
Don quichotte

"Si nous le pendons en effet , il a dit vrai, et ne mérite pas la mort; si nous le laissons passer, il a menti, et la loi veut qu'il soit pendu....
Mais écoutez : quelle que soit notre décision, nous manquerons toujours à la loi;
s'il est pendu, nous sommes en faute, puisqu'il aura dit la vérité;
s'il n'est pas pendu, nous sommes encore en faute, puisqu'il nous aura menti.

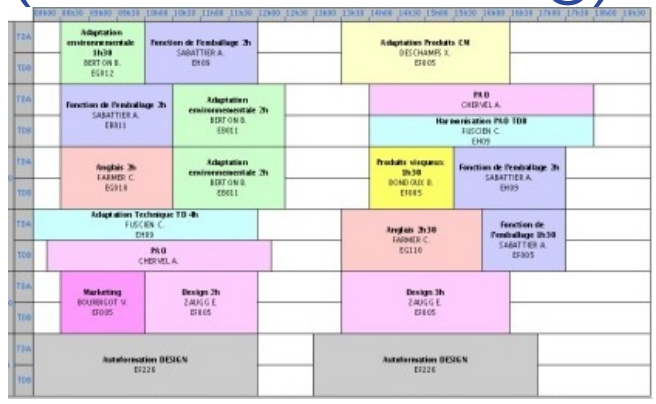
Don quichotte

"Si nous le pendons en effet , il a dit vrai, et ne mérite pas la mort; si nous le laissons passer, il a menti, et la loi veut qu'il soit pendu....
Mais écoutez : quelle que soit notre décision, nous manquerons toujours à la loi;
s'il est pendu, nous sommes en faute, puisqu'il aura dit la vérité;
s'il n'est pas pendu, nous sommes encore en faute, puisqu'il nous aura menti.
Nous n'avons donc que le choix de deux fautes : or, dans ce cas, nous devons choisir celle qui ne fait de mal qu'à nous. Qu'on laisse passer cet homme; s'il aime tant à être pendu , nous le punissons assez en le contrariant pour aujourd'hui."

B-Tree en BD



Conception d'emploi du temps (Timetable desing)



Instance

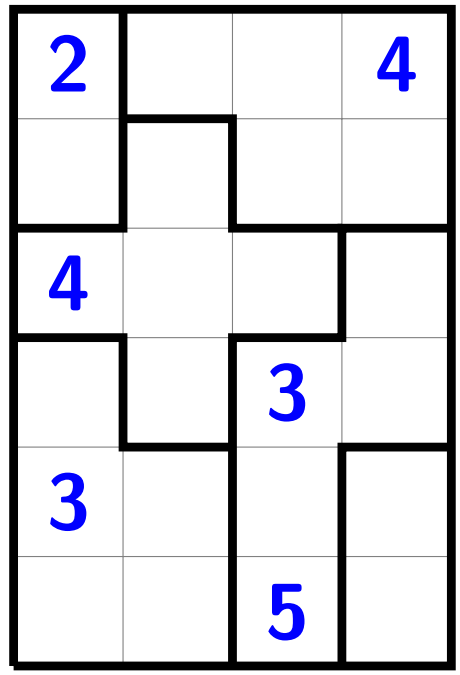
Soit H un ensemble de période de travail, T un ensemble de tâches, C un ensemble d'ouvriers. Soit $A(c) \subseteq H$ les périodes de disponibilités pour chaque ouvriers $c \in C$, soit $A(t) \subseteq H$ les périodes des tâches pour chaque tâche $t \in T$ et pour chaque $(c, t) \in C \times T$ un nombre $R(c, t) \in \mathbb{Z}^{*+}$ de période de travail.

Problème

Existe-il un emploi du temps qui permet de faire toutes les tâches ?
$f : C \times T \times H \to \{0, 1\}$, où $f(c, t, h) = 1$ signifie que c fait la tâche t à l'horaire h.

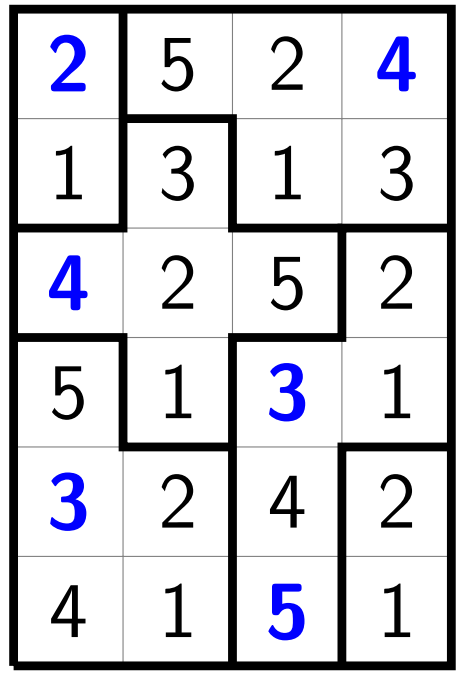
Réduction à 3-SAT par Even, Itai et Shamir 1976.

SUGURU is NP-Complete



- **Number region rule:** A region composed of k cells must be filled with integers $1, \ldots, k$.
- **Neighbour rule:** For every cell, all of its eight neighbours must have different values from the cell's value.

SUGURU solution



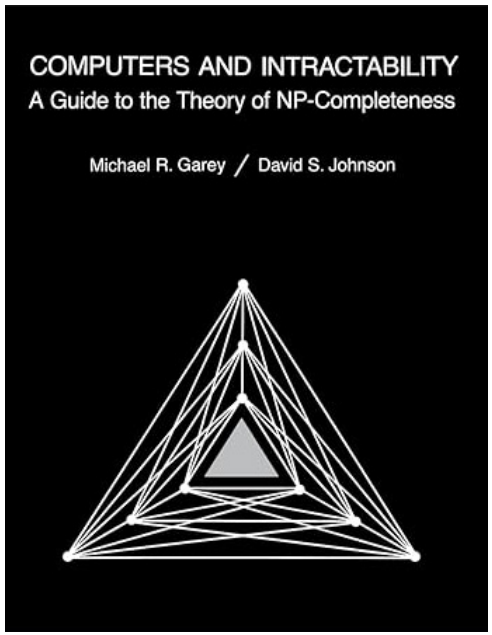
- **Number region rule:** A region composed of k cells must be filled with integers $1, \ldots, k$.
- **Neighbour rule:** For every cell, all of its eight neighbours must have different values from the cell's value.

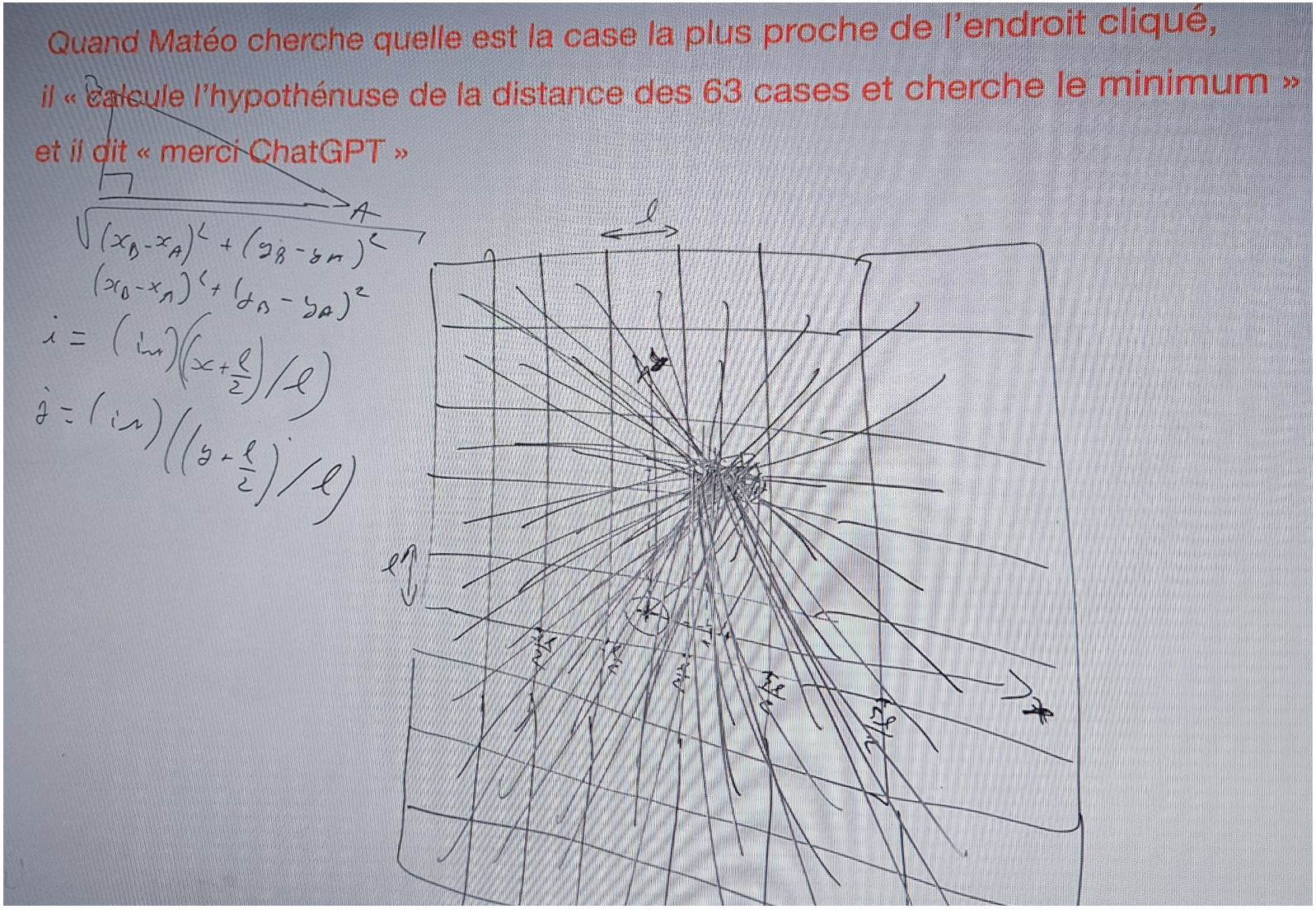
Léo Robert, Daiki Miyahara, Pascal Lafourcade, Luc Libralesso, Takaaki Mizuki 2022

Conclusion

A retenir

- Problèmes ouverts
- Complexité est une science pleine de rebondissement
- Important pour un informaticien
- Théorème de Gödel




Quand Matéo cherche quelle est la case la plus proche de l'endroit cliqué,
il « calcule l'hypothénuse de la distance des 63 cases et cherche le minimum »
et il dit « merci ChatGPT »
A
$\sqrt{(x_B - x_A)^2 + (y_B - y_m)^2}$
$(x_0 - x_n)^2 + (y_n - y_A)^2$
$i = (i_m)((x + \frac{\ell}{2})/\ell)$
$j = (j_m)((y + \frac{\ell}{2})/\ell)$
ℓ
ℓ

Questions?