Calculabilité et Informatique

Jean-Jacques Lévy
INRIA

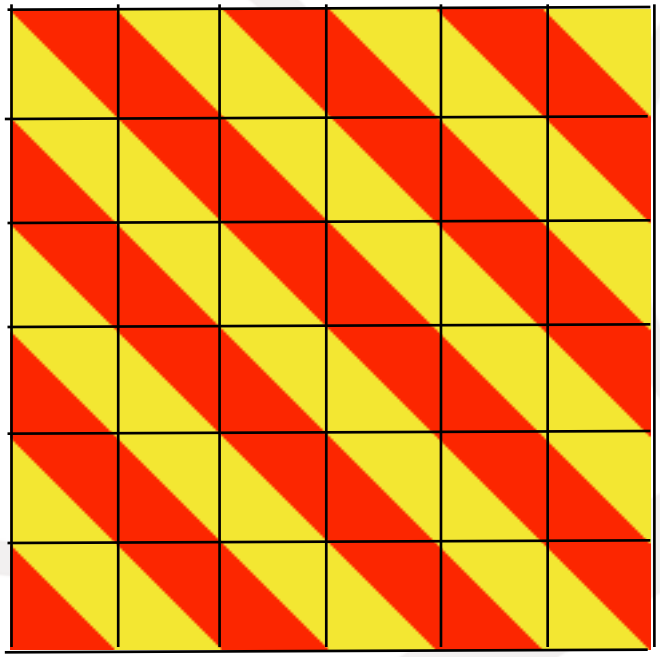
CENTRE DE RECHERCHE COMMUN
INRIA MICROSOFT RESEARCH

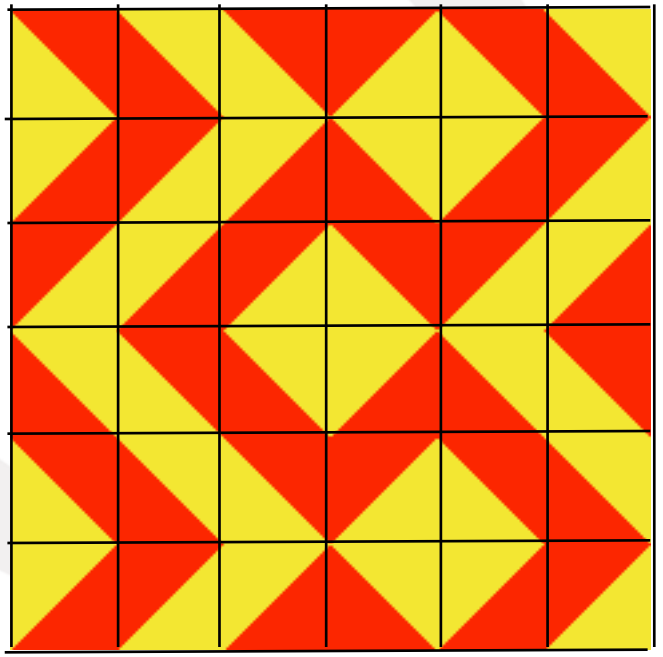
INFORMATIQUE ET RÉALITÉ

DOMINOS
DE WANG

(1950)

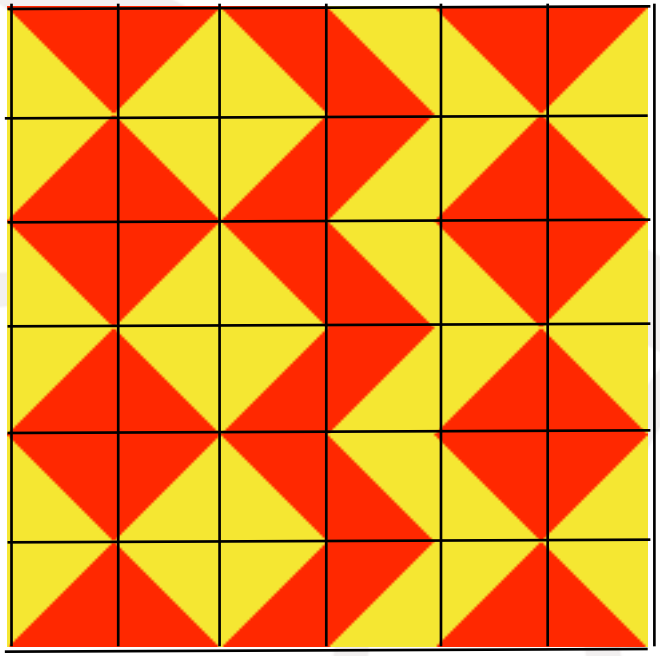
Problème du carreleur

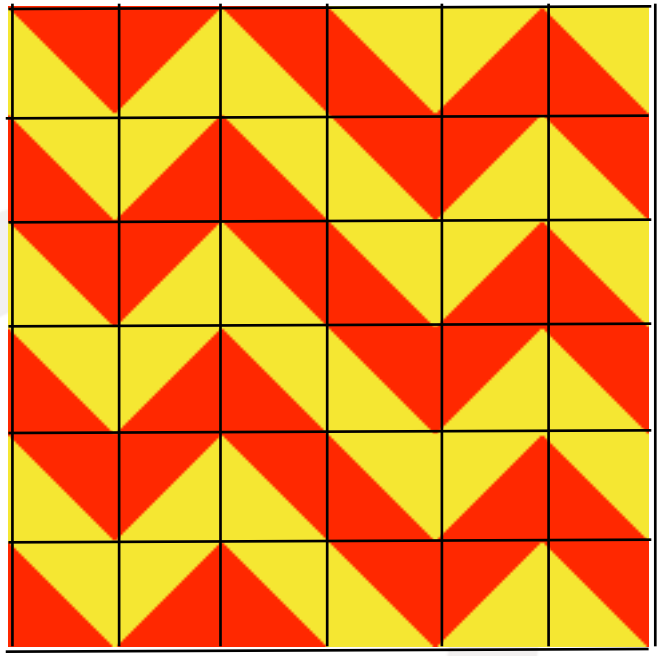


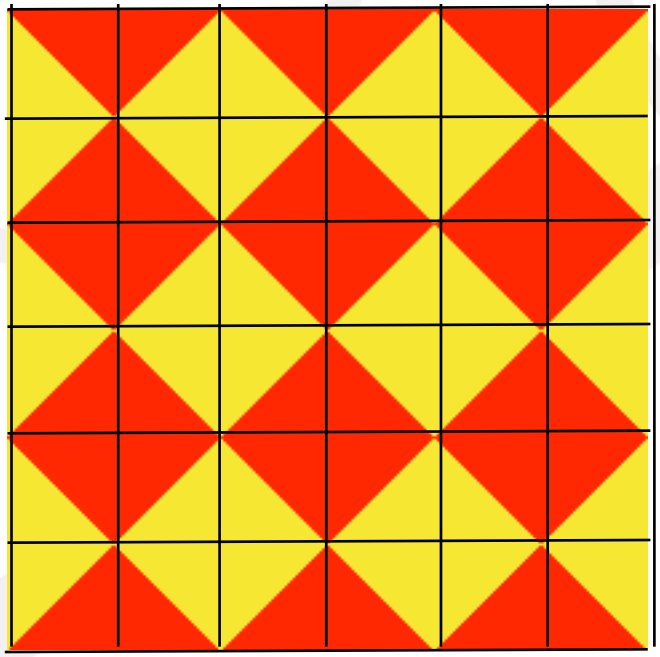


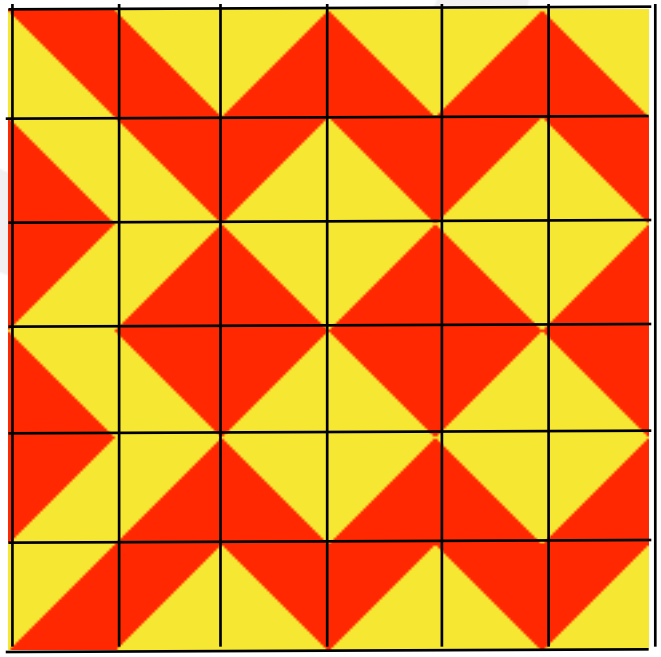
4 motifs

Remplir des carrés de coté 6







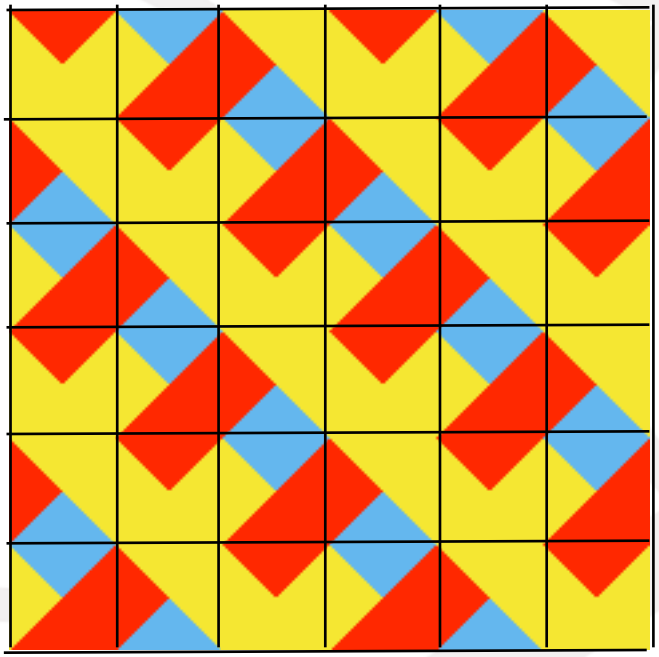


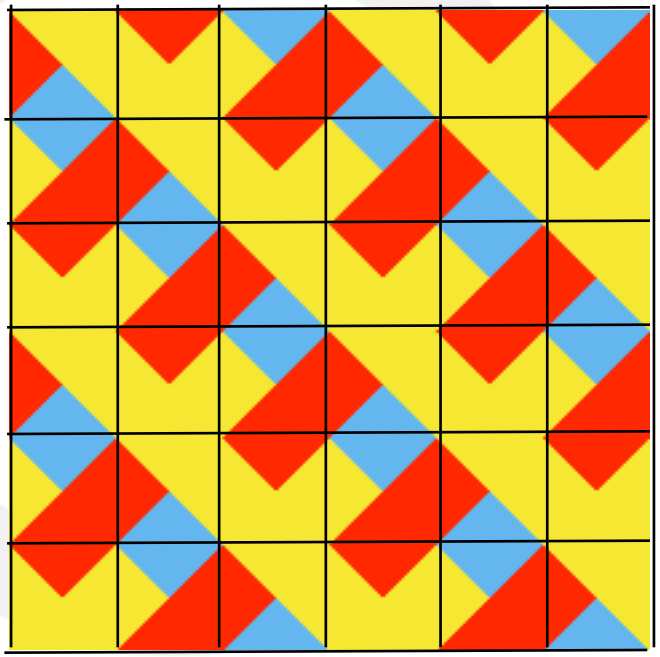
4 motifs

CENTRE DE RECHERCHE COMMUN

INRIA MICROSOFT RESEARCH

Problème du carreleur

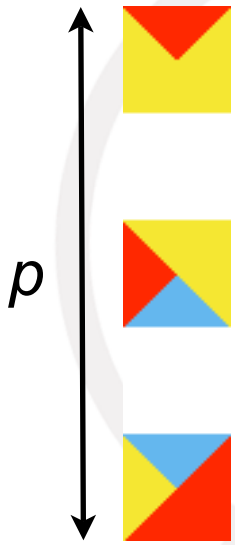


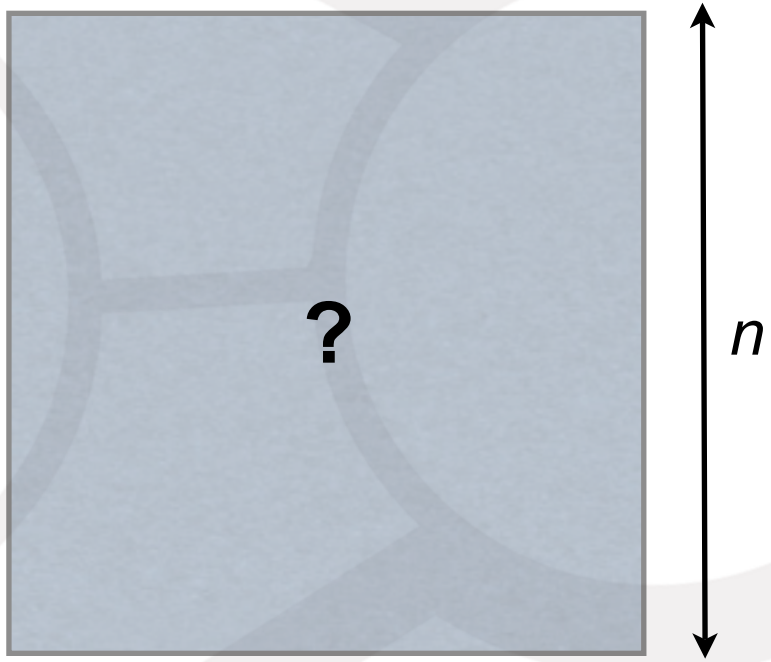


3 motifs

Remplir des carrés de coté 6

Problème du carreleur





p motifs

Remplir des carrés de coté *n*

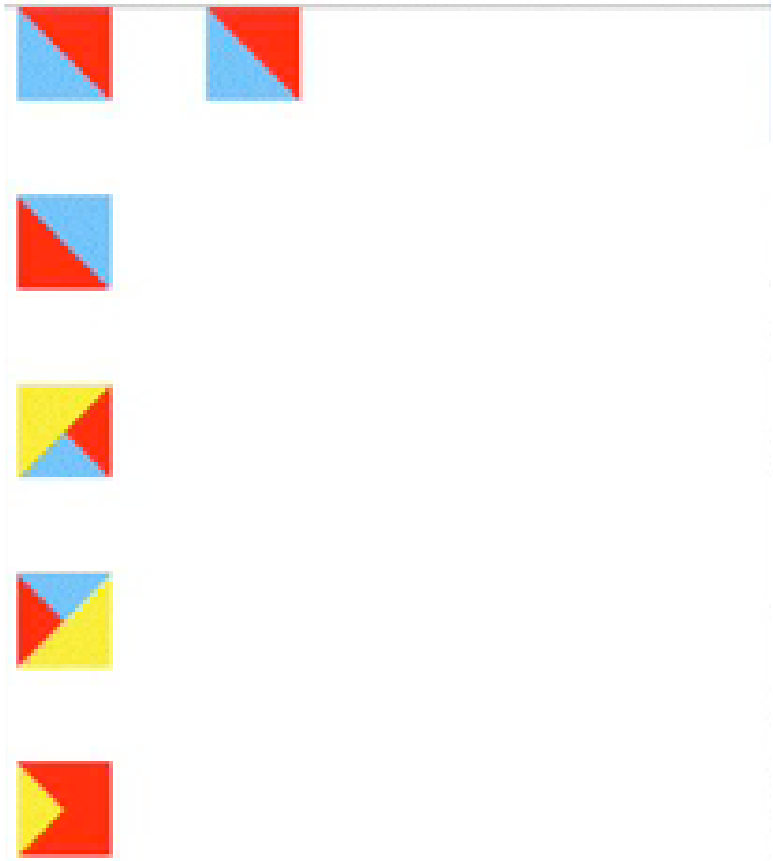
Enumérer tous les carrelages

- Générer tous les carrelages
- Ne retenir que les carrelages licites
- Nombre d'opérations = p^{n^2}
- Si n=6 et p=3, il faut tester 15009463629699121 carrelages
- Soit 57 jours en faisant 500 millions opérations par seconde

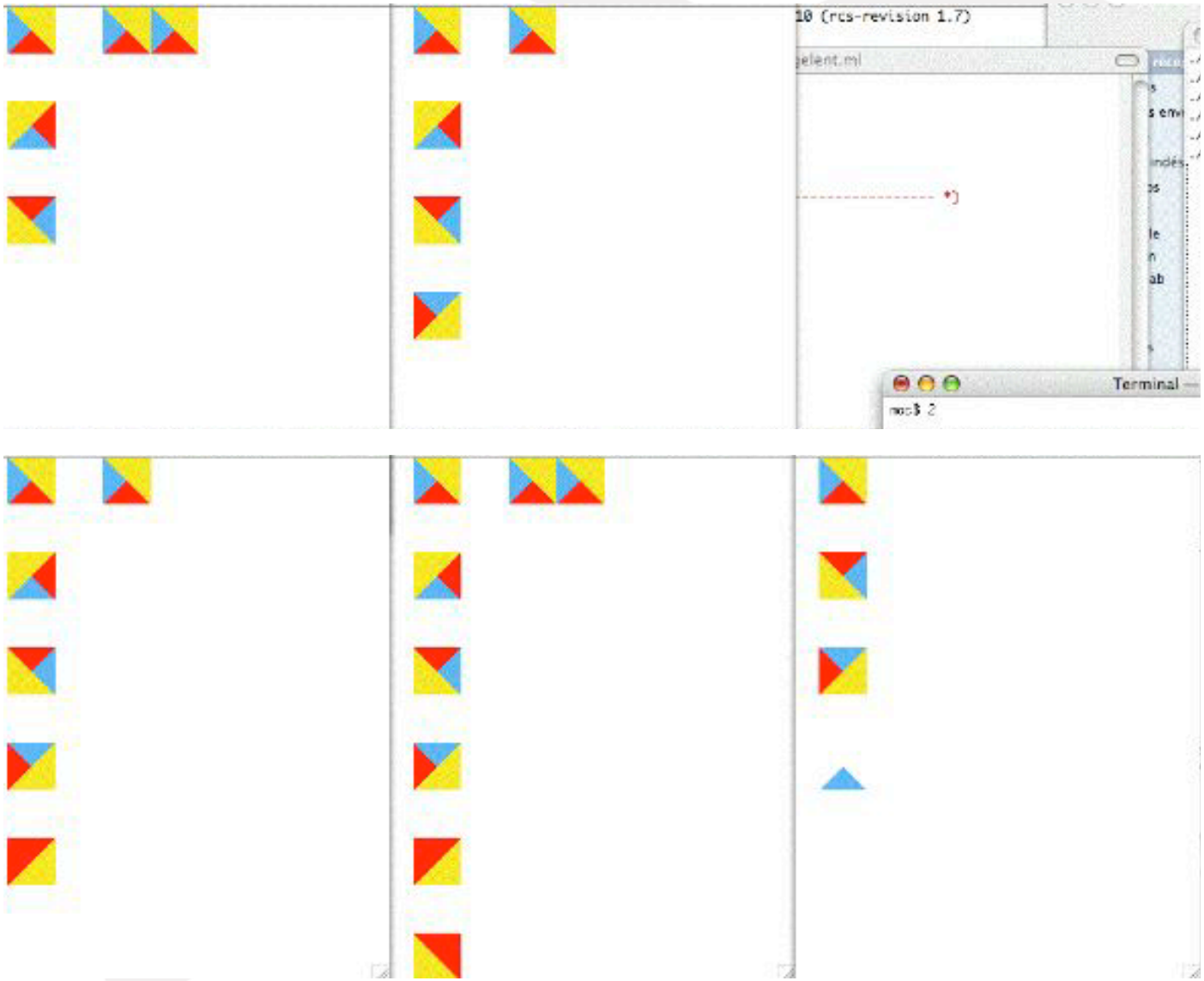
Branch and Bound

- Générer tous les carrelages
- en éliminant les carrelages illicites au fur et à mesure
- Méthode qui marche bien en pratique
- Dans le pire cas, autant d'opérations qu'auparavant

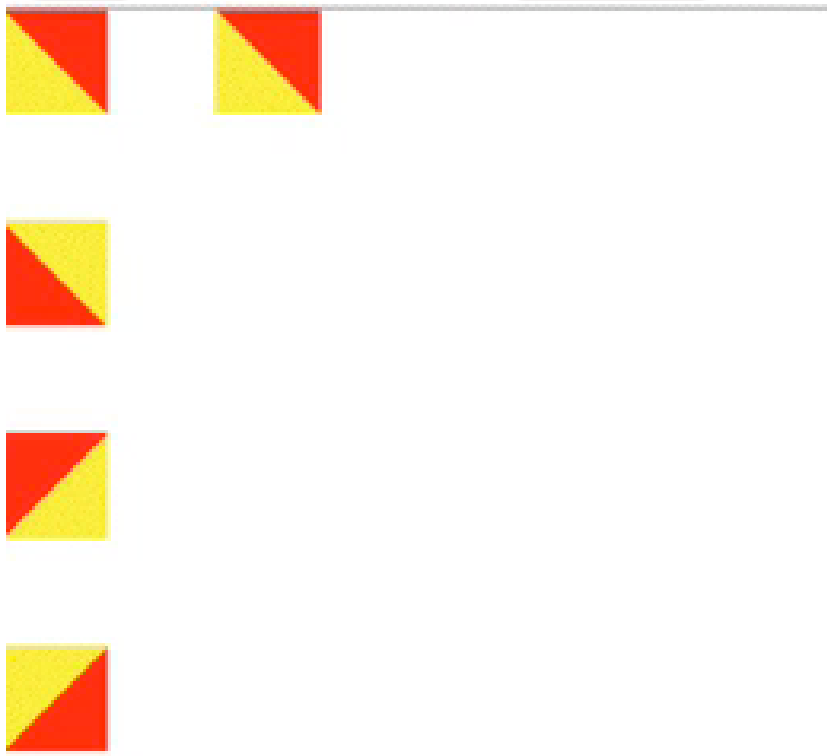
Branch and Bound



Branch and Bound

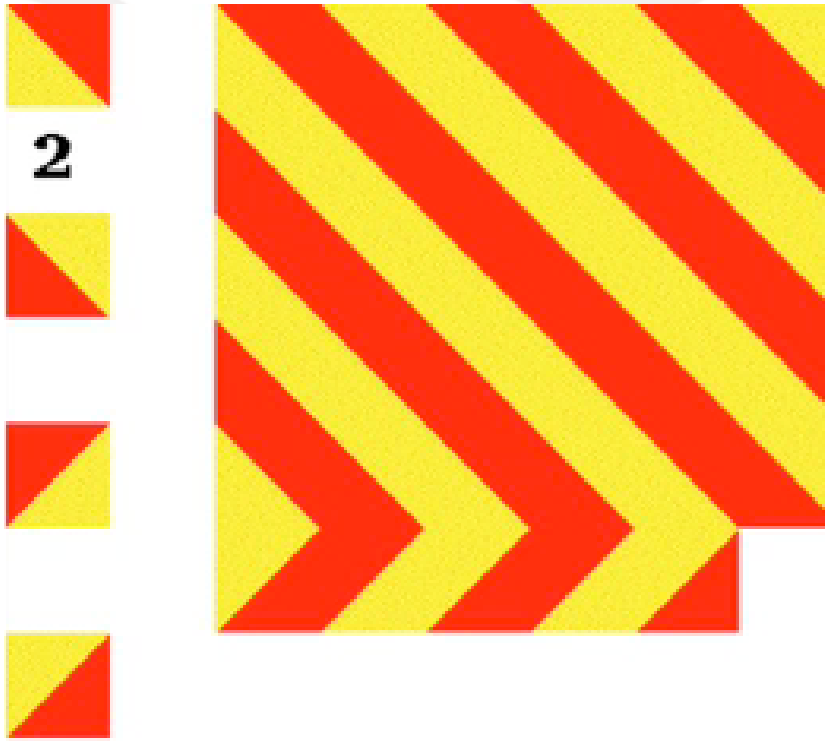


Branch and Bound



Branch and Bound

Branch and Bound



Faire mieux ??

Faire appel à Dieu

- Un oracle indique à chaque étape un bon motif à prendre
- et à la fin, on vérifie si le carrelage est licite
- Méthode très rapide (en temps linéaire et donc polynomial)

Faire appel à Dieu

- Un oracle indique à chaque étape un bon motif à prendre
- et à la fin, on vérifie si le carrelage est licite
- Méthode très rapide (en temps linéaire et donc polynomial)

[On vérifie car Dieu peut se tromper]

Faire appel à Dieu

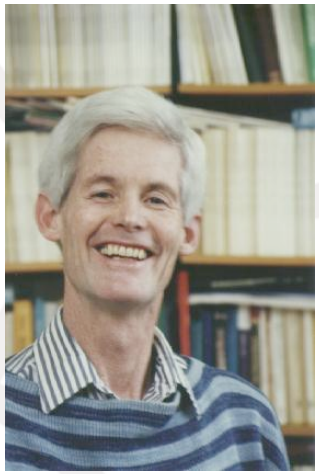
- Un oracle indique à chaque étape un bon motif à prendre
- et à la fin, on vérifie si le carrelage est licite
- Méthode très rapide (en temps linéaire et donc polynomial)

[On vérifie car Dieu peut se tromper]

- mais Dieu peut ne pas exister...

NP complétude

- Personne ne sait faire mieux aujourd'hui
- Ce problème est NP complet.
- Si on sait faire mieux pour le problème du carreleur, on sait le faire pour les nombreux problèmes de la classe NP.



P = NP ?

[Cook, 1973]

- Un des 7 problèmes les plus importants des mathématiques (1000000$ donnés par l'institut Clay)

Carrelage du plan

Le plan est infini

- Donnés: *p* motifs
- But: carreler tout le plan

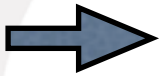
équivalent à [lemme de Koenig]

- Donnés: *p* motifs
- But: carreler tous les carrés de coté *n* pour tout $n \geq 0$

(Si on sait carreler toutes les cuisines du monde, on sait carreler le monde)

Carrelage périodiques

- Wang (1961): tout carrelage du plan est périodique

 il existe un algorithme pour le carrelage du plan

(Le carrelage du plan est décidable)

- Berger (1966) code l'arithmétique avec les dominos de Wang

 il n'existe pas d'algorithme pour le carrelage du plan

Les ordinateurs calculent en binaire

- Une addition

2099	1011
$+\ 1$	$+\ 1$
2100	1100
décimal	binaire

qu'on peut simuler avec les Dominos de Wang

Calculs avec les dominos de Wang

q0	1	0	1	1	B
0	q1	0	1	1	B
0	1	q0	1	1	B
0	1	0	q1	1	B
0	1	0	1	q1	B
0	1	0	1	1	qB
0	1	0	1	r1	B
0	1	0	r1	0	B
0	1	r0	0	0	B
0	1	1	0	0	B

Implementation in Hardware

© Peter van Emde Boas ; 19950310

© Peter van Emde Boas ; 19950310

© Peter van Emde Boas ; 19921031

Le carrelage du plan est indécidable

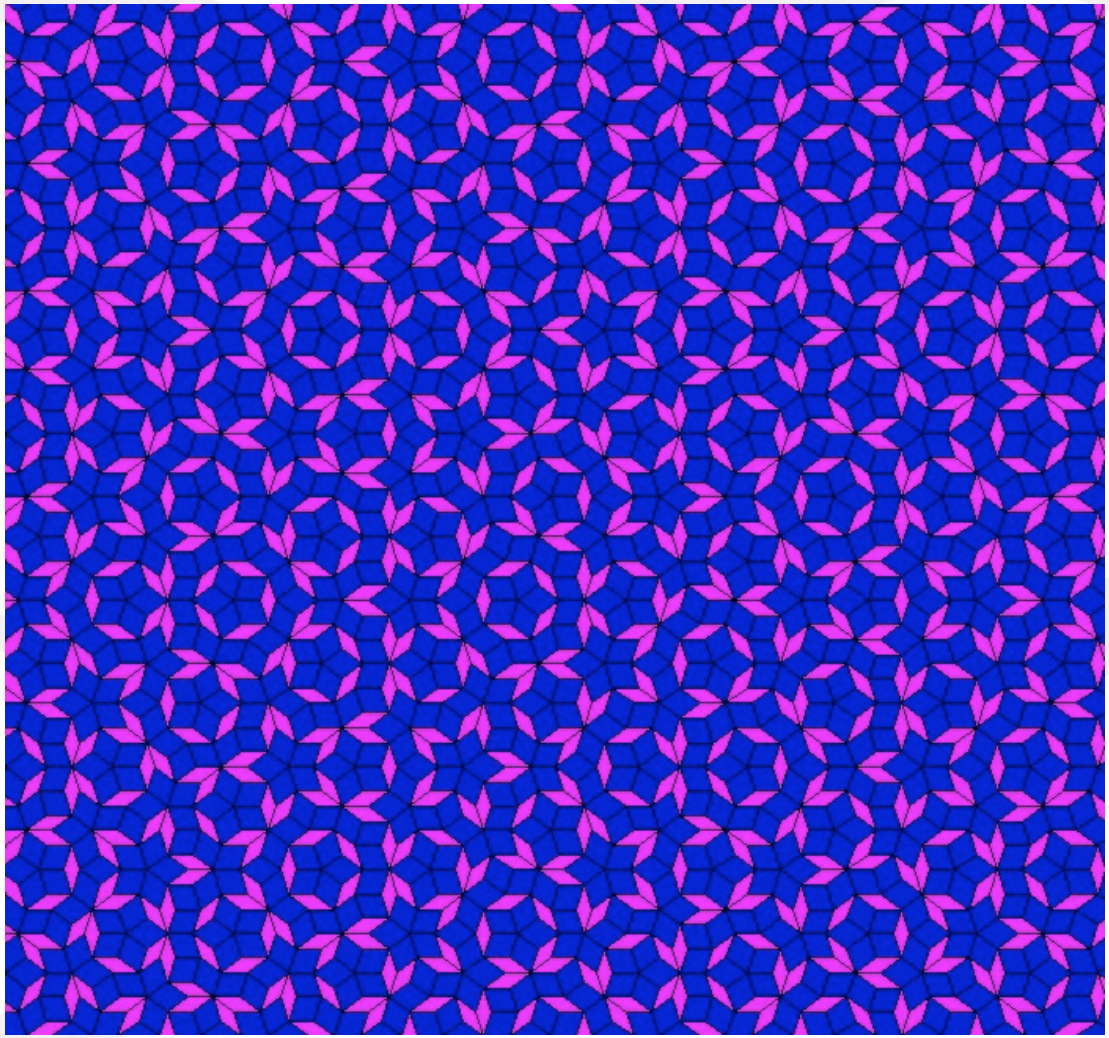
- Indécidabilité du problème de l'arrêt

le carrelage du plan est indécidable

- Il existe donc des carrelages apériodiques du plan

(En 1975, les carrelages apériodiques ont fleuri)

Pavages de Penrose



Conclusion

En informatique, il existe des

- problèmes indécidables
  (carrelage du plan)
- problèmes décidables de complexité élevée
  (carrelage d'un carré)
- problèmes plus faciles en complexité n, n^2, n^3
  (test si un carrelage est licite)

Conclusion

problèmes indécidables

problèmes compliqués

problèmes faciles

LOGIQUE MATHÉMATIQUE

(1910 - 1950)

Les logiciens

"l'ensemble de tous les ensembles"

Les logiciens

"l'ensemble de tous les ensembles"

$$E = \{x \mid x \in x\}$$

Les logiciens

"l'ensemble de tous les ensembles"

$$E = \{x \mid x \in x\}$$

alors $x \in E$ si et seulement si $x \in x$

Les logiciens

"l'ensemble de tous les ensembles"

$$E = \{x \mid x \in x\}$$

alors $x \in E$ si et seulement si $x \in x$

Donc $E \in E$ si et seulement si $E \in E$

Les logiciens

"l'ensemble de tous les ensembles"

$$E = \{x \mid x \in x\}$$

alors $x \in E$ si et seulement si $x \in x$

Donc $E \in E$ si et seulement si $E \in E$

Dilemme!

Les logiciens

"l'ensemble de tous les ensembles"

Les logiciens

"l'ensemble de tous les ensembles"

$$F = \{x \mid x \notin x\}$$

Les logiciens

"l'ensemble de tous les ensembles"

$$F = \{x \mid x \notin x\}$$

alors $x \in F$ si et seulement si $x \notin x$

Les logiciens

"l'ensemble de tous les ensembles"

$$F = \{x \mid x \notin x\}$$

alors $x \in F$ si et seulement si $x \notin x$

Donc $F \in F$ si et seulement si $F \notin F$

Les logiciens

"l'ensemble de tous les ensembles"

$$F = \{x \mid x \notin x\}$$

alors $x \in F$ si et seulement si $x \notin x$

Donc $F \in F$ si et seulement si $F \notin F$

Paradoxe!

Les logiciens

``je dis que je mens''

Les logiciens

"je dis que je mens"

- si je mens en disant que je mens, je dis donc la vérité.

Les logiciens

``je dis que je mens''

- si je mens en disant que je mens, je dis donc la vérité.

- mais si je dis la vérité et dis que je mens, je mens et ne raconte donc pas la vérité.

Les logiciens

``je dis que je mens''

- si je mens en disant que je mens, je dis donc la vérité.

- mais si je dis la vérité et dis que je mens, je mens et ne raconte donc pas la vérité.

Paradoxe du menteur!

Les logiciens

"je dis que je mens"

- si je mens en disant que je mens, je dis donc la vérité.

- mais si je dis la vérité et dis que je mens, je mens et ne raconte donc pas la vérité.

Paradoxe du menteur!

On est souvent incohérent quand on parle de soi-même ("réfléxivité").

Les logiciens

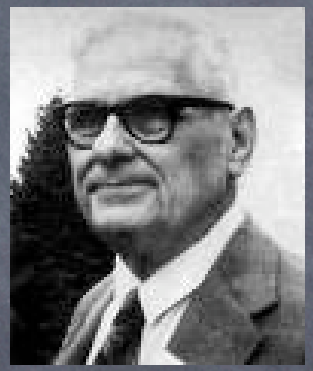
Hilbert → Gödel → Church → Turing

Church → Kleene

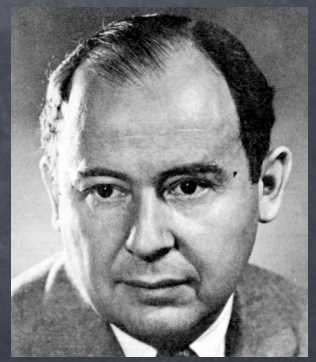
Kleene → Post

Kleene → von Neumann









Les logiciens

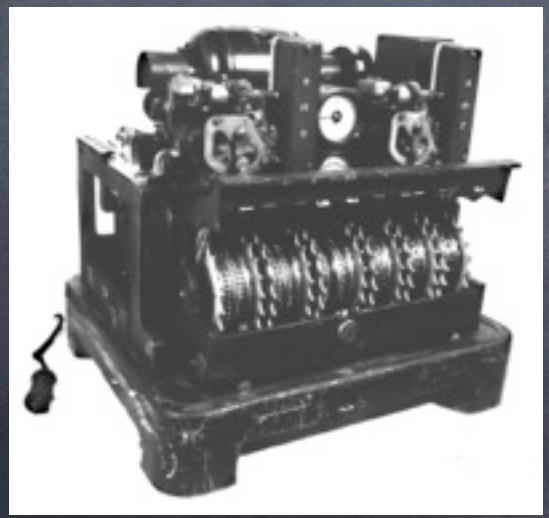
- Turing (décodage de la machine Enigma)
 contrôle fini, mémoire infinie

- von Neumann (projet Manhattan)
 données ET programmes en mémoire

- machines de UPenn, Cambridge, Mark I

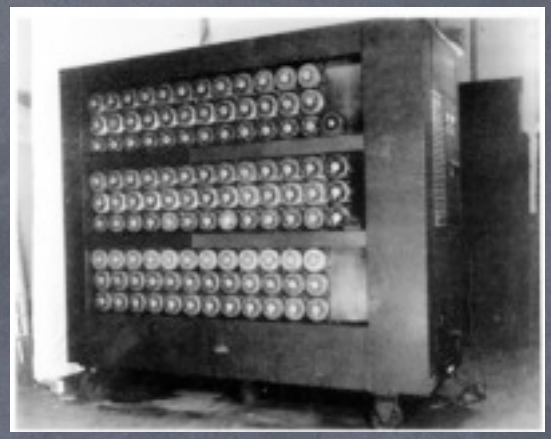
Allemagne

Enigma



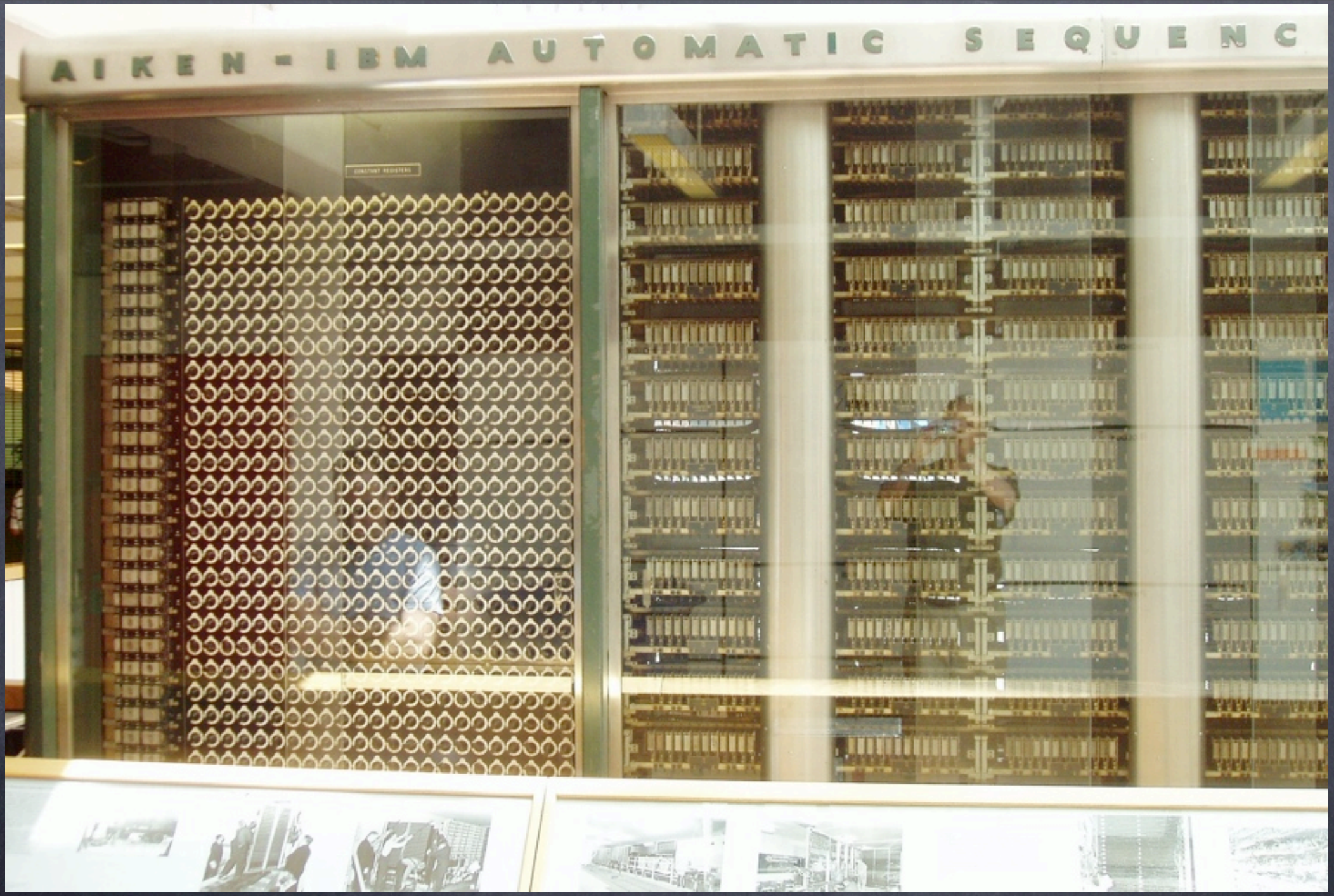
Lorenz

Bletchley Park

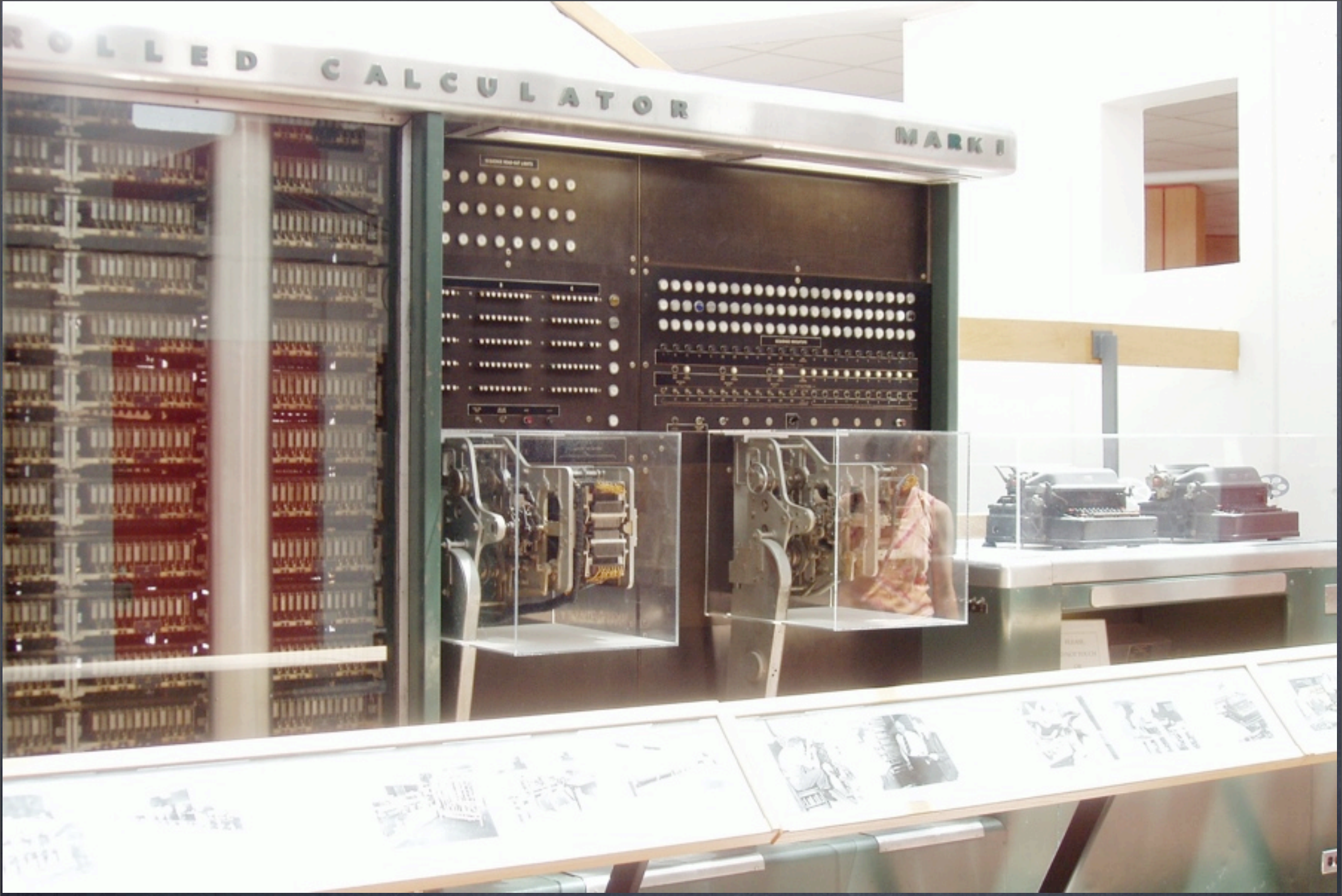


The Bombe

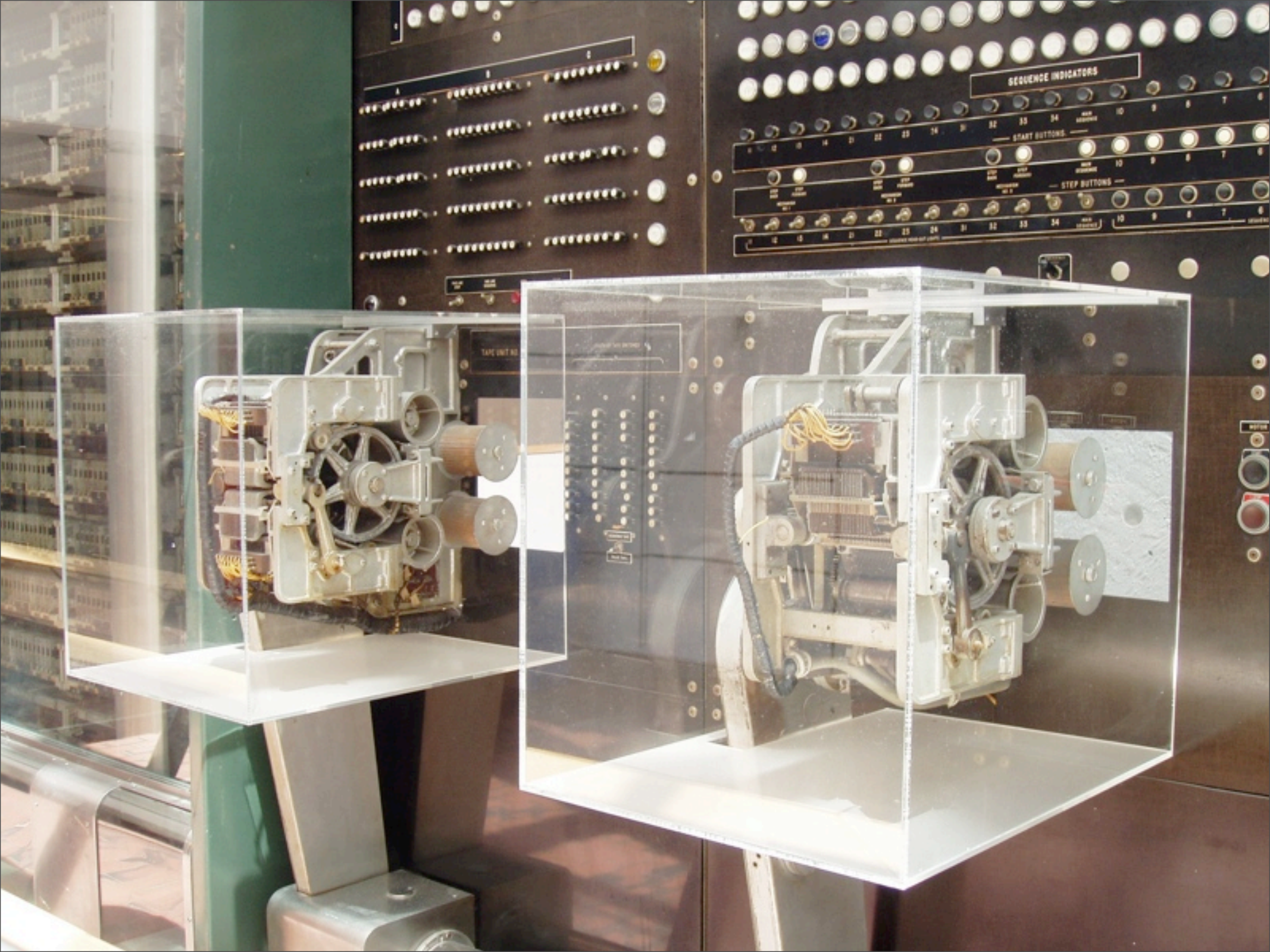
Colossus



Mark I, Harvard



Mark I, Harvard


SEQUENCE INDICATORS
START BUTTONS
STEP BUTTONS

Les logiciens

Mark II,
Harvard

92

9/9

0800 Arctan started { 1.2700 9.037 847 025
1000 " stopped - arctan ✓ 9.037 846 995 correct
13" wc (032) MP - MC ~~1.9821447000~~ ~~2.13047764715 (033)~~ 4.615925059(-2)
(033) PRO 2 2.13047764715
correct 2.13067764715

Relays 6-2 in 033 failed special speed test
In relay " " 11.00 test.
Relays changed

Relay 2145
Relay 3370

1100 Started Cosine Tape (Sine check)
1525 Started Multy Adder Test.

1545

Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

~~1525~~ 1630 Arctangent started.
1700 closed down.

MACHINES INFORMATIQUES

(1950-1980)

Multics

- temps partagé
- 10 à 100 utilisateurs / ordinateur
- courrier électronique

- IBM 704, 360/370
- SDS 940, Butler Lampson
- GE 645, Multics; MIT, Bull

Unix, nirvanha des informaticiens

- simplification de Multics
- modularité "small is beautiful"

- AT&T Bell laboratories
- théoriciens ET praticiens

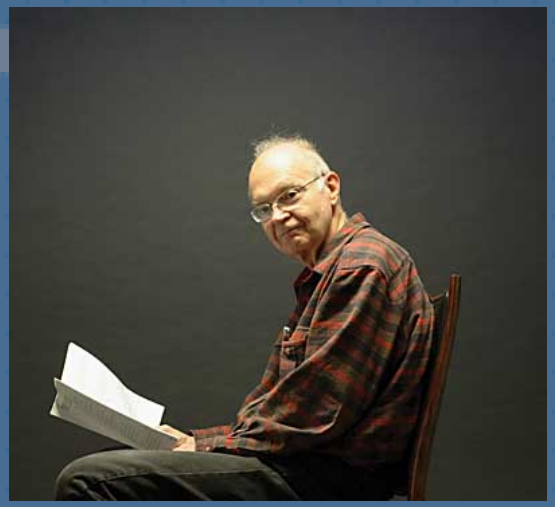
- système de hackers pour hackers
- pdp 11; Vax 780/750

Faire fonctionner les ordinateurs

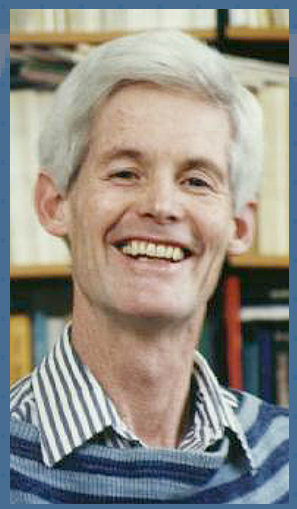
- langages de programmation
- systèmes d'exploitation

Jean Ichbiah

- correction des programmes
- trouver de bons algorithmes



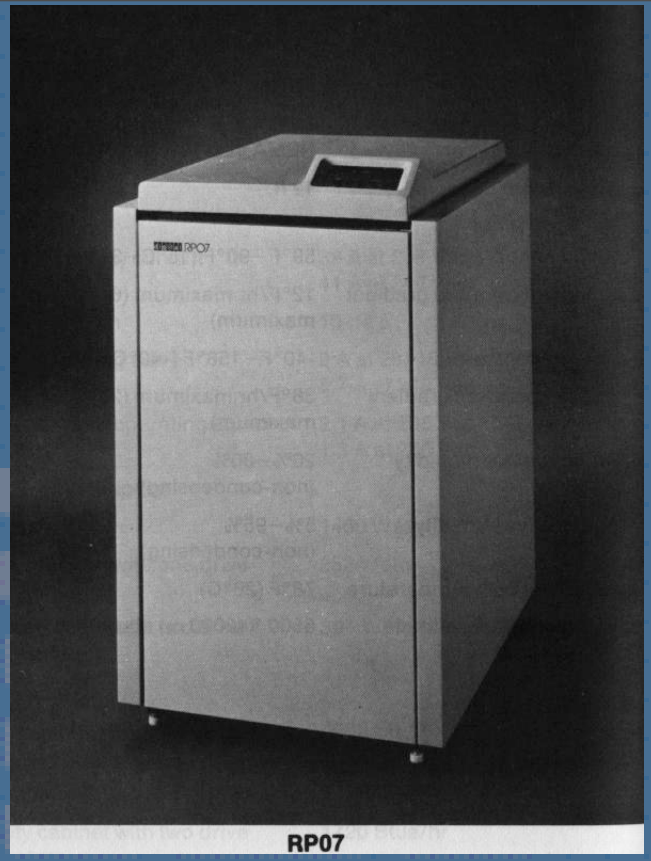
Don Knuth



Steve Cook

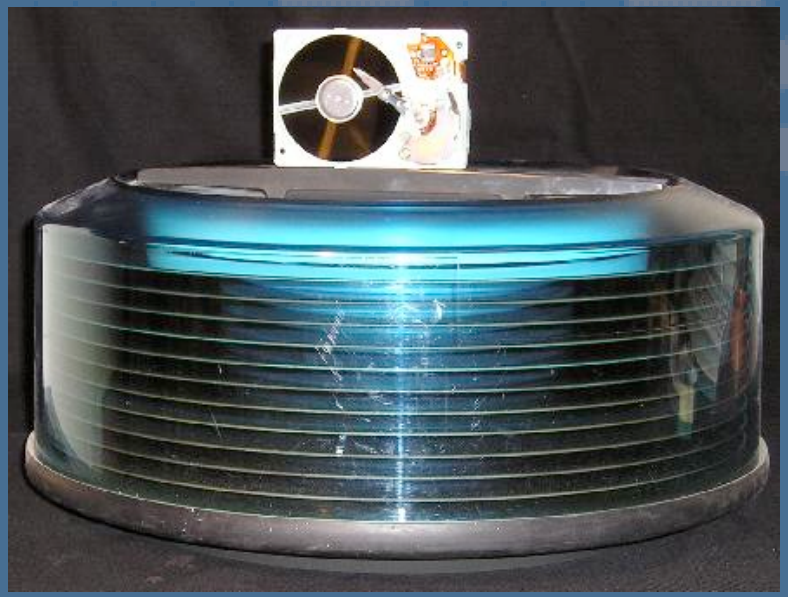
P=NP ?

vax 11/750

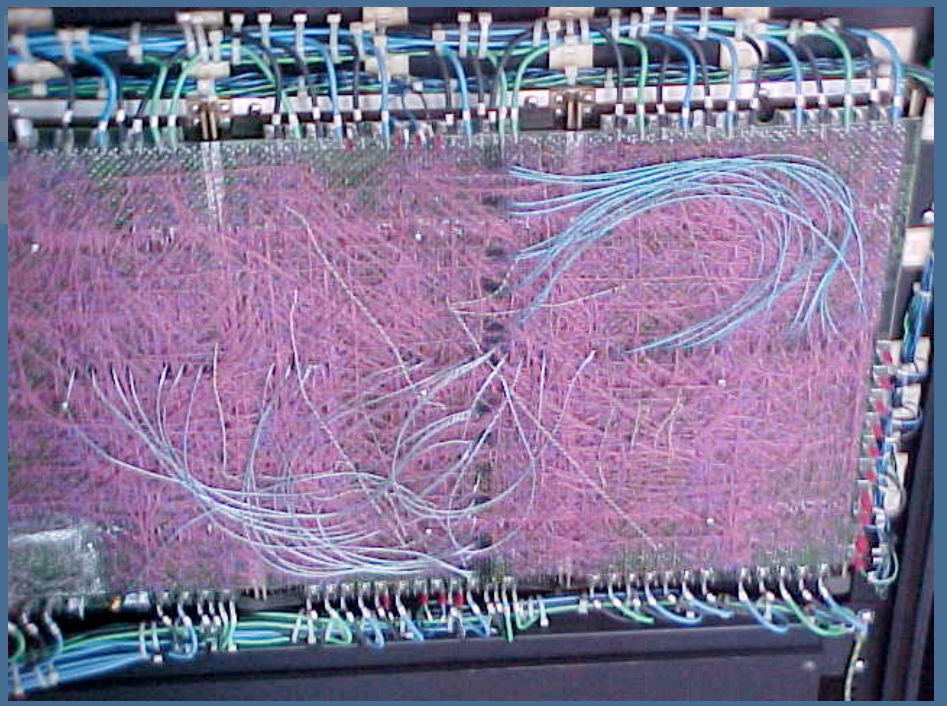


RP07

RP07 (700MO)



RM05 (256 MO)



arrière d'un dec 10

vax 11/780

MACHINES PERSONNELLES

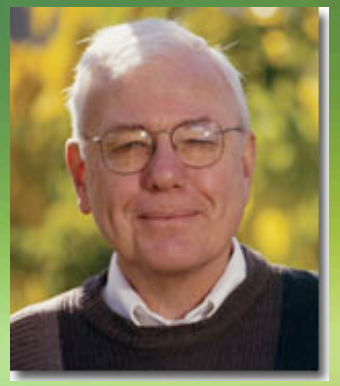
(1980-2000)

Le garage

- intel 4004
- Xerox PARC (alto, dorado)
- le garage Apple (apple II, lisa, macintosh)
- IBM PC (ms-dos)

Alto

- vision égoiste
- tout le monde a son ordinateur
- seul l'interface compte



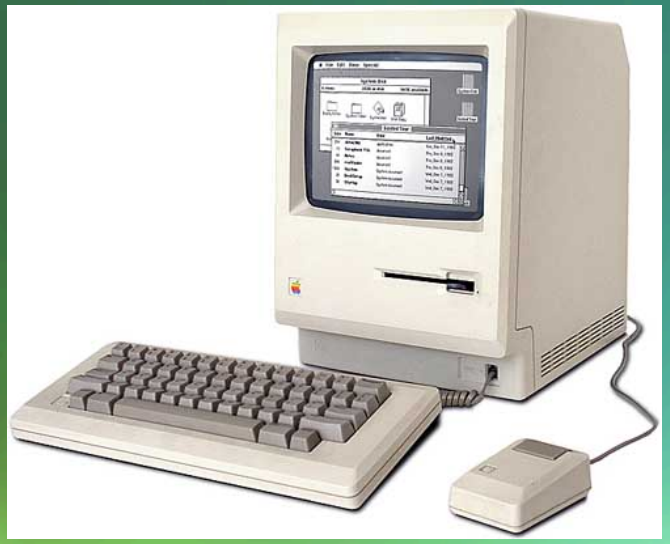
Chuck Thacker

Apple II

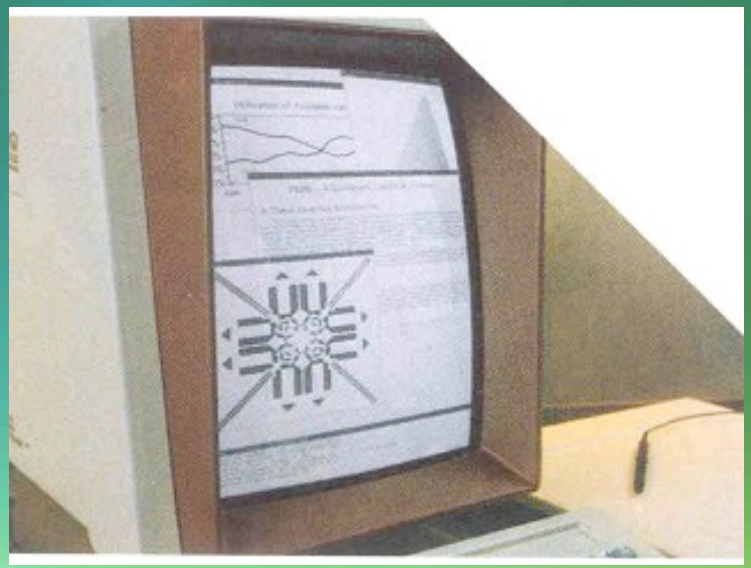
Lisa

Apollo

Sun 1



Macintosh



PerQ

Blit 5620

Microsoft

- ordinateur dans chaque maison
- bureautique (Word, Excel, Powerpoint)
- éditeur de logiciel, pas de matériel
- améliorations du système (NT, 95, XP, Vista)

Charles Simonyi

- Dave Cutler (DEC-VMS, NT)
- éditeurs WYSIWYG (bravo, Word)

Linux et le logiciel libre

- Emacs, éditeur de texte extensible
- gcc, compilateur C de la Free Software Foundation
- Linux = Unix refait par Linus Tordsvald
- tout le monde participe au système
- source public mais invasif
- logiciels de qualité

Richard Stallman

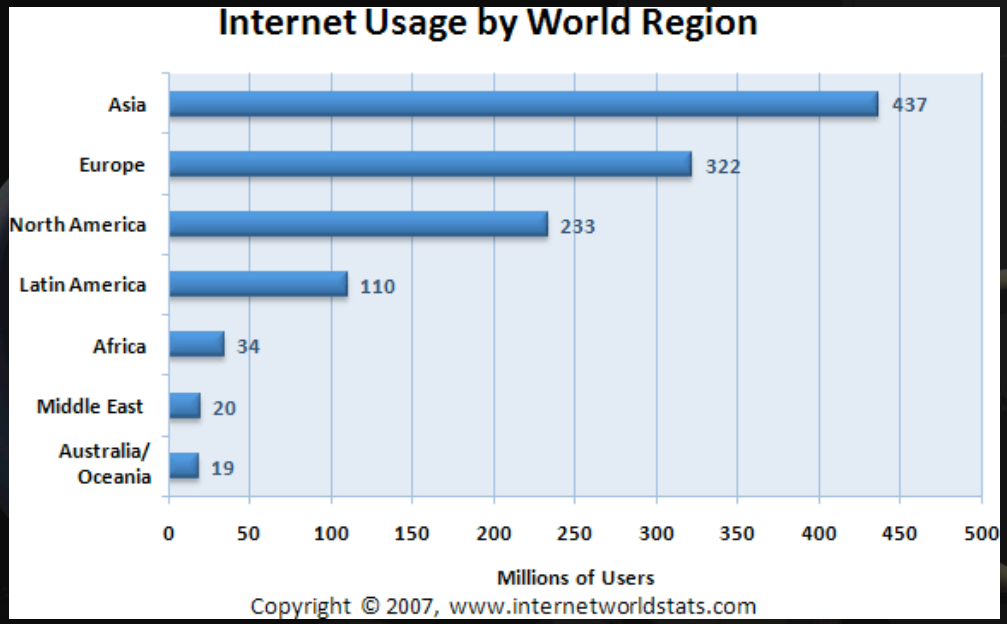
INTERNET, L'ORDINATEUR MONDIAL

(2000-2007)

Internet

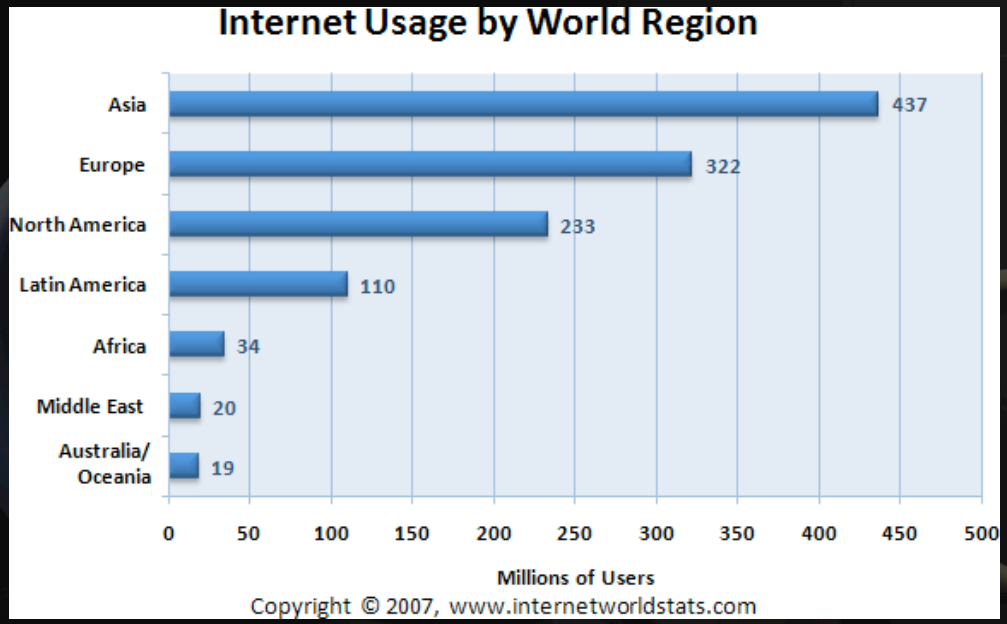
- arpanet (1970), ethernet - cyclades (1975)
- uucp (1985), internet (1992), wifi (1999)
- 1 milliard d'internautes (2007)
- web 30%, p2p 30%, mail 2% du traffic
- 25% du téléphone sur IP

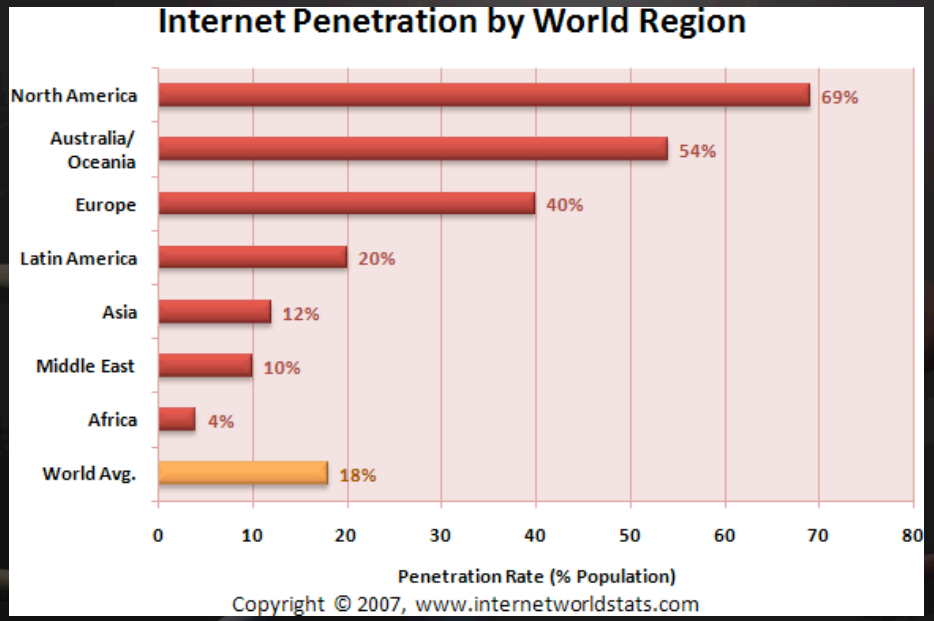
Internet



- 33 M internautes en France

Internet



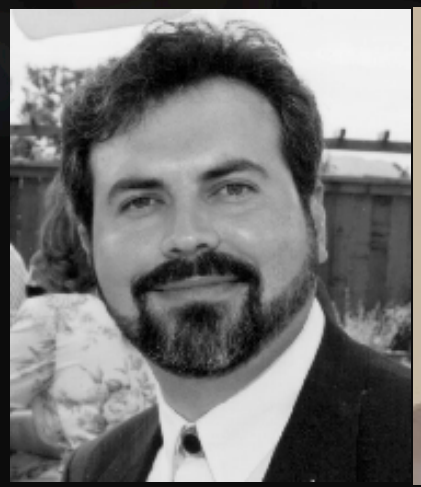


- 33 M internautes en France

Internet

- informatique ubiquitaire
- le réseau est la propriété de tous
- les données ne sont plus localisées
- rôle des indexeurs (altavista, google, ...)

Louis Monier



Mike Burrows

Google

- recherche globale
- a embauché l'équipe Unix de Bell labs
- 12 centres contenant les données mondiales
- services réseaux (courrier, calendrier, cartes)
- eBay, amazon, skype (2003), youtube (2005)

Réseaux et distribution

- sécurité (secret, authentification, intégrité)
- agencement réparti des données
- programmation distribuée
- jeux
- capteurs

FUTUR

(2007-???)

Futur?

- disparition des ordinateurs
- développement des capteurs
- médecine et informatique
- programmation des cellules biologiques
- applications à l'environnement
- ordinateurs quantiques
- ???

CENTRE DE RECHERCHE COMMUN

INRIA
MICROSOFT RESEARCH