La Nuit du Quantique

De l'informatique classique à l'informatique quantique

Pourquoi faire le pas ?

Bogdan Vulpescu

Laboratoire de Physique de Clermont Auvergne

Service Informatique

31 mars 2026

Préambule

L’informatique (classique) : **la technologie de l’information**, IT = *Information Technology* [EN]

Préambule

L’informatique (classique) : **la technologie de l’information**, IT = *Information Technology* [EN]

- calcul avec des nombres, algorithmes de calcul, résultats des calculs

Préambule

L’informatique (classique) : **la technologie de l’information**, IT = *Information Technology* [EN]

- calcul avec des nombres, algorithmes de calcul, résultats des calculs

- organisation des données (formats de fichiers : .txt, .doc, .xls, .odp, .jpg, .csv, .npy, .bin, …)

Préambule

L’informatique (classique) : **la technologie de l’information**, IT = *Information Technology* [EN]

- calcul avec des nombres, algorithmes de calcul, résultats des calculs
- organisation des données (formats de fichiers : .txt, .doc, .xls, .odp, .jpg, .csv, .npy, .bin, …)
- stockage des données, base de données, fouille de données, tri des données et autres manipulations

Préambule

L’informatique (classique) : **la technologie de l’information**, IT = *Information Technology* [EN]

- calcul avec des nombres, algorithmes de calcul, résultats des calculs
- organisation des données (formats de fichiers : .txt, .doc, .xls, .odp, .jpg, .csv, .npy, .bin, …)
- stockage des données, base de données, fouille de données, tri des données et autres manipulations
- encodage / décodage des données, compression, protocoles de transmission des données (applications de la théorie de l’information), internet

Préambule

L’informatique (classique) : **la technologie de l’information**, IT = *Information Technology* [EN]

- calcul avec des nombres, algorithmes de calcul, résultats des calculs
- organisation des données (formats de fichiers : .txt, .doc, .xls, .odp, .jpg, .csv, .npy, .bin, …)
- stockage des données, base de données, fouille de données, tri des données et autres manipulations
- encodage / décodage des données, compression, protocoles de transmission des données (applications de la théorie de l’information), internet
- cryptographie, protection des données, confidentialité des données, communication

Faire des calculs avec les nombres

Faire des calculs avec les nombres

objet ⇒ élément d'un ensemble ⇒ unité ⇒ 1 + 1 + 1 + 1 + 1 + 1 + … **N fois**

Faire des calculs avec les nombres

objet ⇒ élément d'un ensemble ⇒ unité ⇒ 1 + 1 + 1 + 1 + 1 + 1 + … **N fois**

⇒ (1 + 1 + 1 + … + 1) 2314 fois = 2 fois **mille** + 3 fois **cent** + 1 fois **dix** + 4 fois **un**

Faire des calculs avec les nombres

objet ⇒ élément d'un ensemble ⇒ unité ⇒ 1 + 1 + 1 + 1 + 1 + 1 + … **N fois**

⇒ (1 + 1 + 1 + … + 1) 2314 fois = 2 fois **mille** + 3 fois **cent** + 1 fois **dix** + 4 fois **un**

En **base de numération 10** : $2314 = 2 \cdot 10^3 + 3 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0 = (2314)_{10}$

Faire des calculs avec les nombres

objet ⇒ élément d'un ensemble ⇒ unité ⇒ 1 + 1 + 1 + 1 + 1 + 1 + … **N fois**

⇒ (1 + 1 + 1 + … + 1) 2314 fois = 2 fois **mille** + 3 fois **cent** + 1 fois **dix** + 4 fois **un**

En **base de numération 10** : $2314 = 2 \cdot 10^3 + 3 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0 = (2314)_{10}$

nombres fractionnaires : $\begin{cases} 0.2314 = 0.2000 + 0.0300 + 0.0010 + 0.0004 \\ 0.2314 = 2 \cdot 10^{-1} + 3 \cdot 10^{-2} + 1 \cdot 10^{-3} + 4 \cdot 10^{-4} \end{cases}$

Faire des calculs avec les nombres

objet ⇒ élément d'un ensemble ⇒ unité ⇒ 1 + 1 + 1 + 1 + 1 + 1 + … **N fois**

⇒ (1 + 1 + 1 + … + 1) 2314 fois = 2 fois **mille** + 3 fois **cent** + 1 fois **dix** + 4 fois **un**

En **base de numération 10** : $2314 = 2 \cdot 10^3 + 3 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0 = (2314)_{10}$

nombres fractionnaires : $\begin{cases} 0.2314 = 0.2000 + 0.0300 + 0.0010 + 0.0004 \\ 0.2314 = 2 \cdot 10^{-1} + 3 \cdot 10^{-2} + 1 \cdot 10^{-3} + 4 \cdot 10^{-4} \end{cases}$

La **base de numération** utilisée l'ordinateur classique est **2** : 10 ➔ 2 et {0, 1, 2, … , 9} ➔ {0, 1}

Faire des calculs avec les nombres

objet ⇒ élément d'un ensemble ⇒ unité ⇒ 1 + 1 + 1 + 1 + 1 + 1 + … **N fois**

⇒ (1 + 1 + 1 + … + 1) 2314 fois = 2 fois **mille** + 3 fois **cent** + 1 fois **dix** + 4 fois **un**

En **base de numération 10** : $2314 = 2 \cdot 10^3 + 3 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0 = (2314)_{10}$

nombres fractionnaires : $\begin{cases} 0.2314 = 0.2000 + 0.0300 + 0.0010 + 0.0004 \\ 0.2314 = 2 \cdot 10^{-1} + 3 \cdot 10^{-2} + 1 \cdot 10^{-3} + 4 \cdot 10^{-4} \end{cases}$

La **base de numération** utilisée l'ordinateur classique est **2** : 10 ➔ 2 et {0, 1, 2, … , 9} ➔ {0, 1}

$$(2314)_{10} = (100100001010)_2 = 1 \cdot 2^{11} + 1 \cdot 2^8 + 1 \cdot 2^3 + 1 \cdot 2^1 = 2048 + 256 + 8 + 2$$

$$\Rightarrow \quad (0.2314)_{10} = (0.0011101100111101)_2 = 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-7} + 1 \cdot 2^{-8} + 1 \cdot 2^{-11} + 1 \cdot 2^{-12} + 1 \cdot 2^{-13} + 1 \cdot 2^{-14} + 1 \cdot 2^{-16}$$

Sources :
https://www.rapidtables.com/convert/number/decimal-to-binary.html?x=2314

Le format en virgule flottante selon le standard international **IEEE-754**

Le format en virgule flottante selon le standard international **IEEE-754**

Simple précision

⇒ sur 32 bits (ou 4 octets)

Le format en virgule flottante selon le standard international **IEEE-754**

Simple précision

⇒ sur 32 bits (ou 4 octets)

Exemple :

0.0345 ⇒ **0 01111010 00011010100111111011111 = 3.4499999e-2 = 0.034499999**

Le format en virgule flottante selon le standard international **IEEE-754**

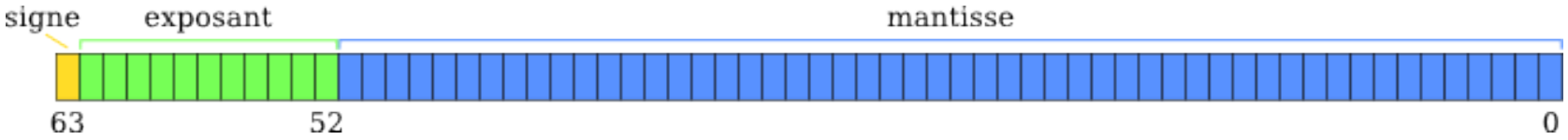
Simple précision

⇒ sur 32 bits (ou 4 octets)

Exemple :

0.0345 ⇒ **`0 01111010 00011010100111111011111` = 3.4499999e-2 = 0.034499999**

et avec **double précision** (sur 64 bits, ou 8 octets)



Le format en virgule flottante selon le standard international **IEEE-754**

Simple précision

⇒ sur 32 bits (ou 4 octets)

Exemple :

0.0345 ⇒ **0 01111010 00011010100111111011111 = 3.4499999e-2 = 0.034499999**

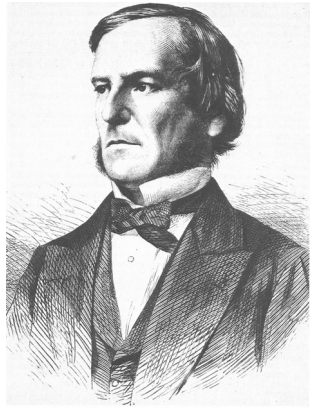
et avec **double précision** (sur 64 bits, ou 8 octets)

0.0345 ⇒ **0 01111111010**
0001101010011111101111100111011011001000101101000100
= 3.450000000000000e-2 = 0.0345

Sources :
IEEE-754 ⇒ https://christophervickery.com/babbage/IEEE-754/IEEE-754.old/Decimal.html
Images Par GMjeanmatt — Travail personnel, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7318466
IEEE = Institute of Electrical and Electronics Engineers

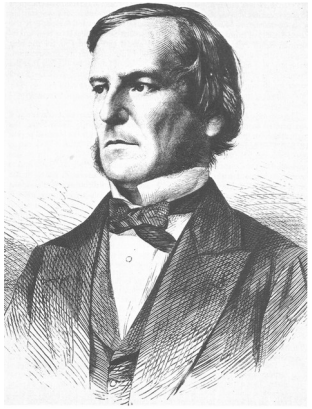
Une algèbre spéciale : l’algèbre Booléenne

George Boole
1815-1864
Source :
Wikipedia

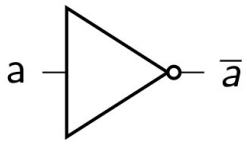


Une algèbre spéciale : l'algèbre Booléenne

George Boole
1815-1864
Source :
Wikipedia



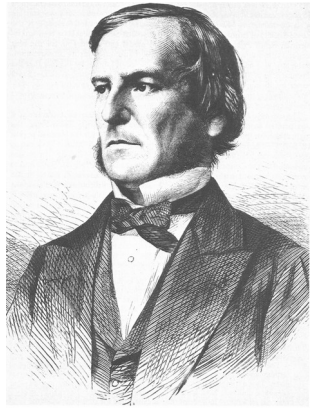
⇒ NOT



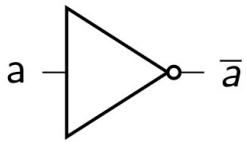
a	$\overline{a}$
0	1
1	0

Une algèbre spéciale : l'algèbre Booléenne

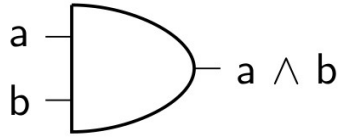
George Boole
1815-1864
Source :
Wikipedia



⇒ NOT, AND



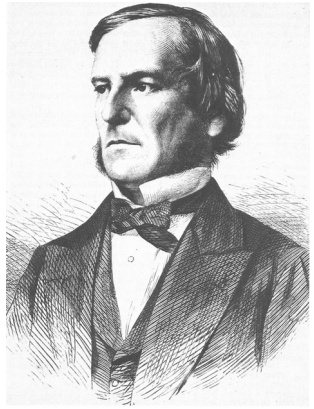
a	$\overline{a}$
0	1
1	0



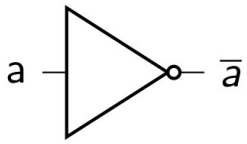
a	b	a ∧ b
0	0	0
0	1	0
1	0	0
1	1	1

Une algèbre spéciale : l'algèbre Booléenne

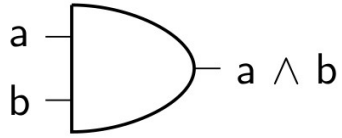
George Boole
1815-1864
Source :
Wikipedia



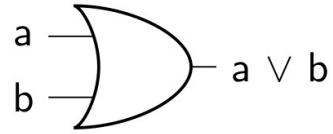
⇒ NOT, AND, OR



a	$\overline{a}$
0	1
1	0



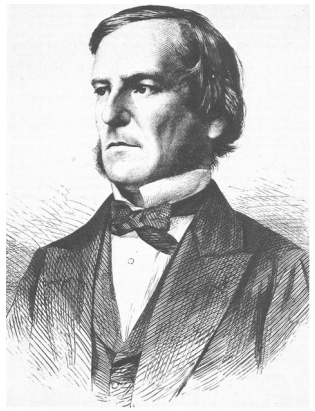
a	b	a ∧ b
0	0	0
0	1	0
1	0	0
1	1	1



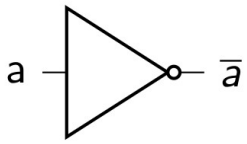
a	b	a ∨ b
0	0	0
0	1	1
1	0	1
1	1	1

Une algèbre spéciale : l'algèbre Booléenne

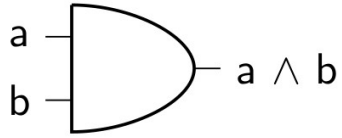
George Boole
1815-1864
Source :
Wikipedia



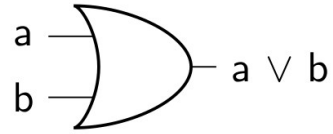
⇒ NOT, AND, OR, XOR



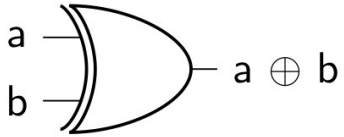
a	$\overline{a}$
0	1
1	0



a	b	a ∧ b
0	0	0
0	1	0
1	0	0
1	1	1



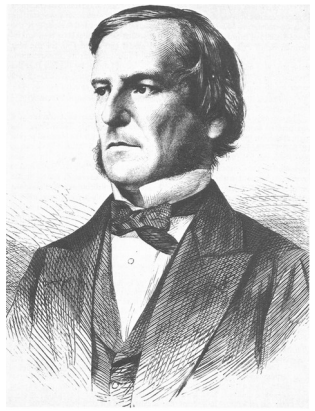
a	b	a ∨ b
0	0	0
0	1	1
1	0	1
1	1	1



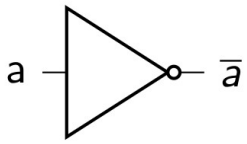
a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0

Une algèbre spéciale : l'algèbre Booléenne

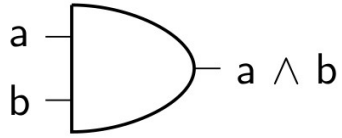
George Boole
1815-1864
Source :
Wikipedia



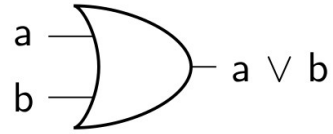
⇒ NOT, AND, OR, XOR, NAND



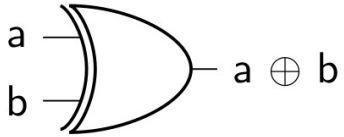
a	$\overline{a}$
0	1
1	0



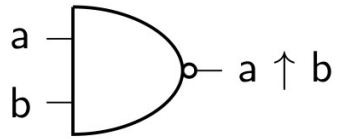
a	b	a ∧ b
0	0	0
0	1	0
1	0	0
1	1	1



a	b	a ∨ b
0	0	0
0	1	1
1	0	1
1	1	1



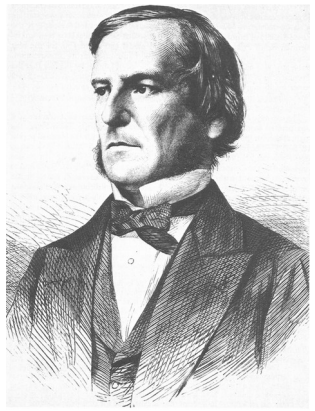
a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0



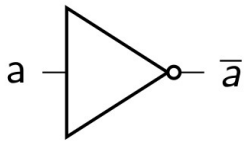
a	b	a ↑ b
0	0	1
0	1	1
1	0	1
1	1	0

Une algèbre spéciale : l'algèbre Booléenne

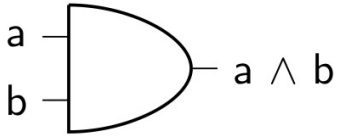
George Boole
1815-1864
Source :
Wikipedia



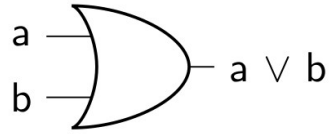
⇒ NOT, AND, OR, XOR, NAND, NOR : tables de vérité et diagrammes de circuit



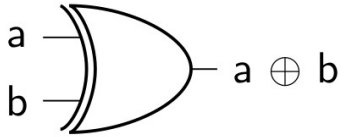
a	$\overline{a}$
0	1
1	0



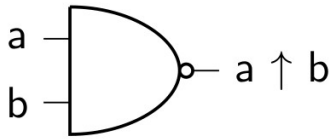
a	b	a ∧ b
0	0	0
0	1	0
1	0	0
1	1	1



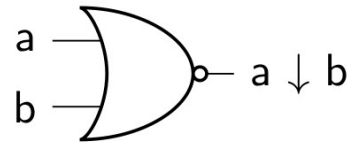
a	b	a ∨ b
0	0	0
0	1	1
1	0	1
1	1	1



a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0

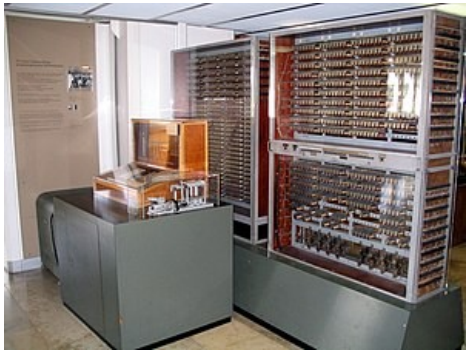


a	b	a ↑ b
0	0	1
0	1	1
1	0	1
1	1	0

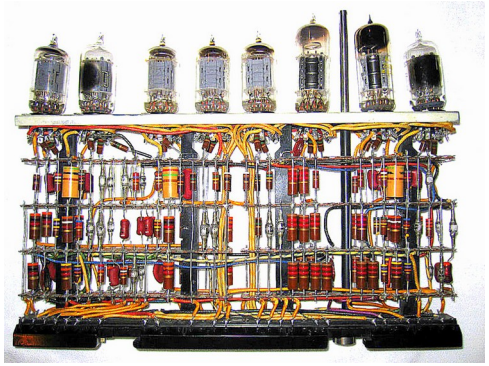


a	b	a ↓ b
0	0	1
0	1	0
1	0	0
1	1	0

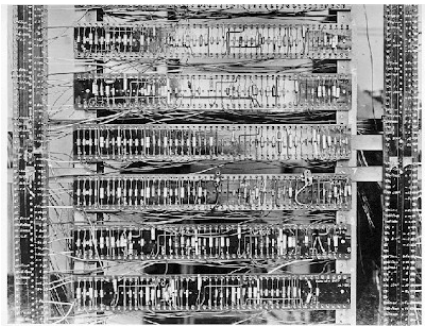
Le (micro-)processeur classique



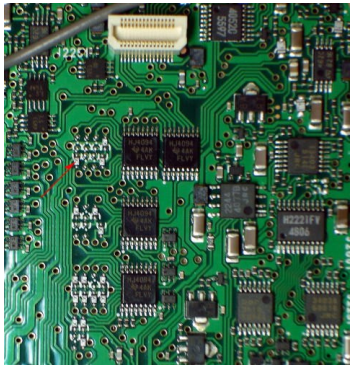
électro-mécanique



tubes

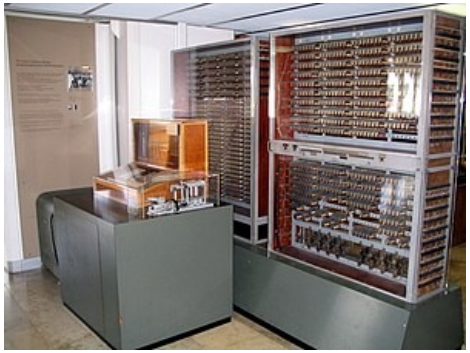


transistors

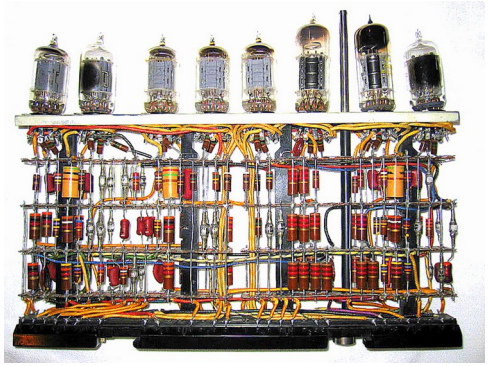


circuits intégrés

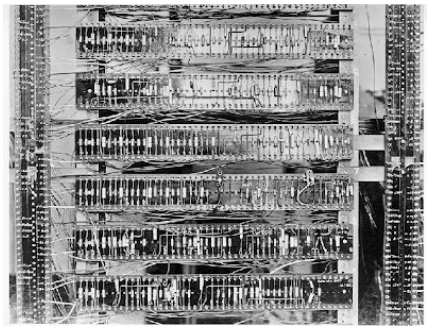
Le (micro-)processeur classique



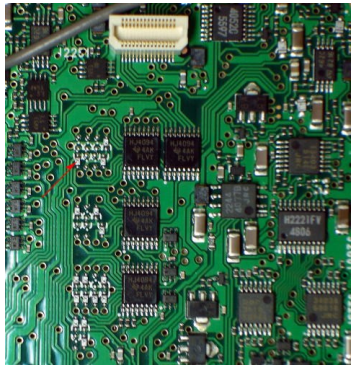
électro-mécanique



tubes

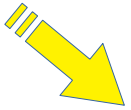


transistors

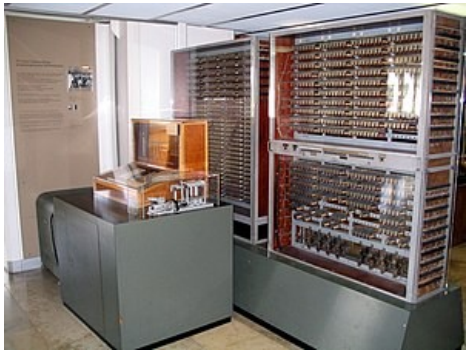


circuits intégrés

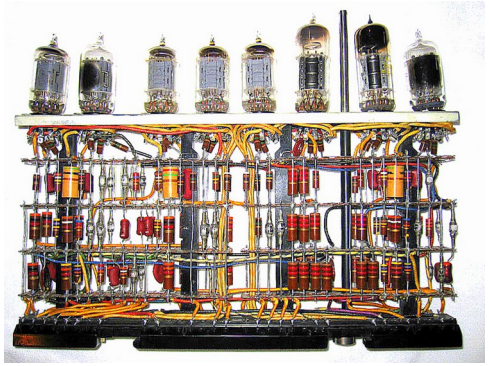
dimensions……..



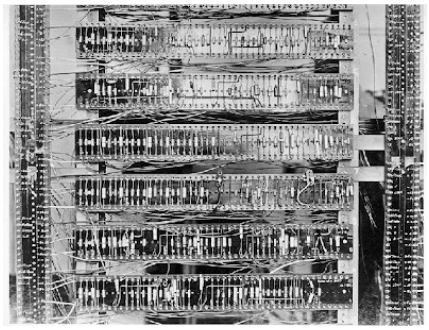
Le (micro-)processeur classique



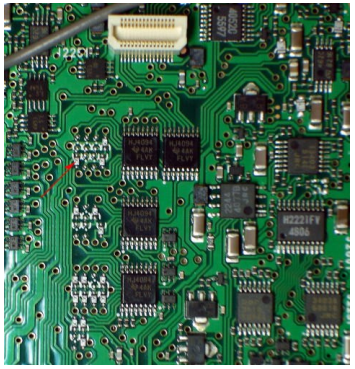
électro-mécanique



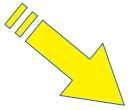
tubes

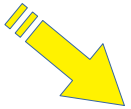


transistors

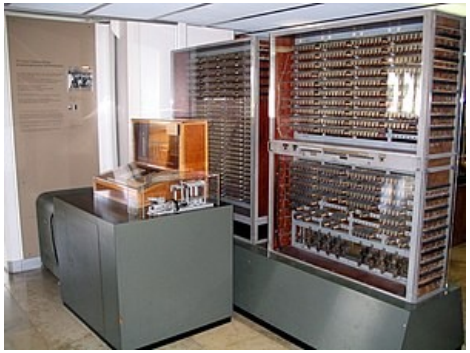


circuits intégrés

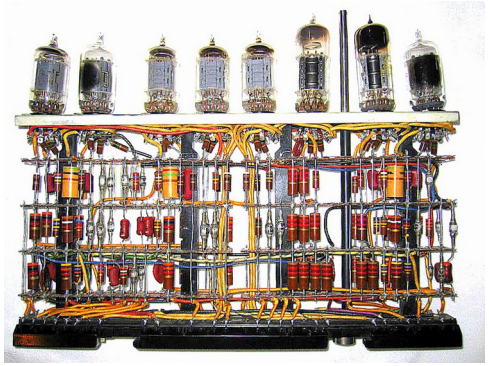
dimensions……..... 

consommation électrique... 

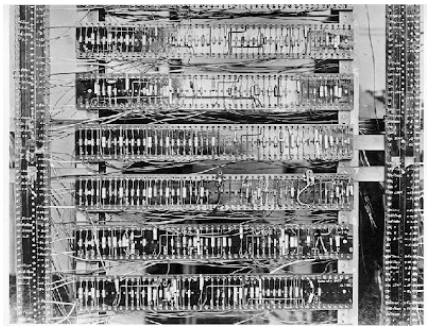
Le (micro-)processeur classique



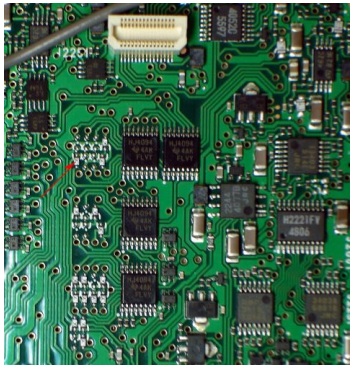
électro-mécanique



tubes



transistors



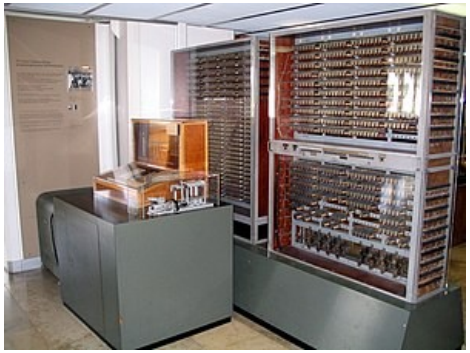
circuits intégrés

dimensions……..

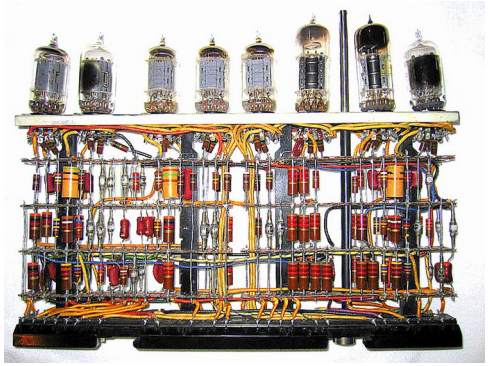
consommation électrique...

vitesse de calcul……..

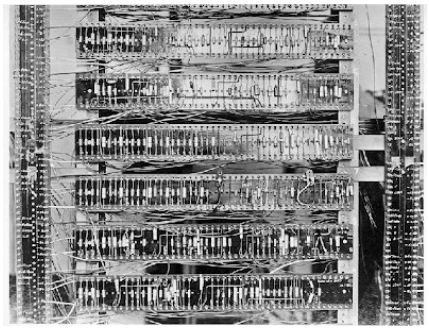
Le (micro-)processeur classique



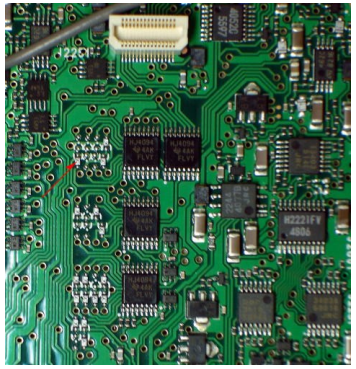
électro-mécanique



tubes



transistors



circuits intégrés

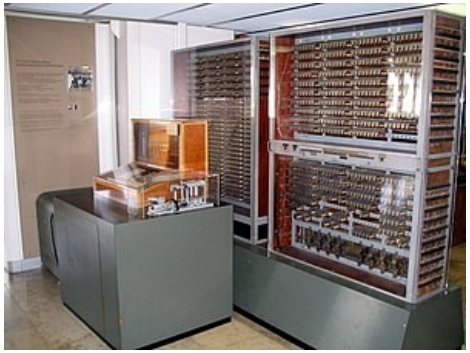
dimensions……..

consommation électrique...

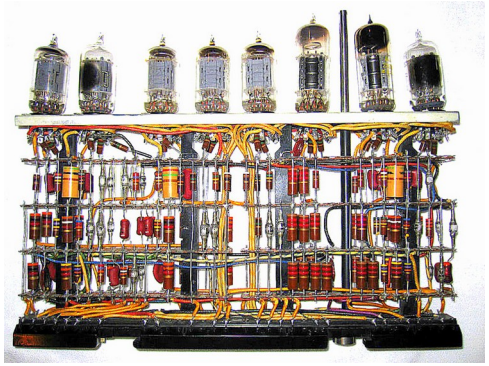
vitesse de calcul……..

capacité de calcul…….

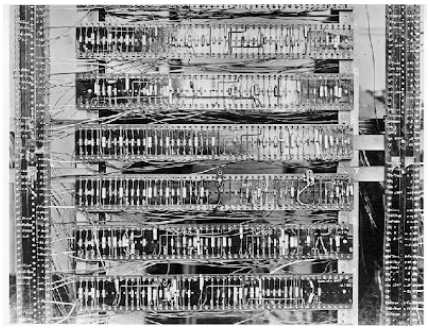
Le (micro-)processeur classique



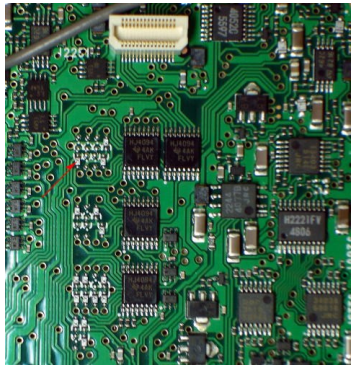
électro-mécanique



tubes



transistors



circuits intégrés

dimensions……..

consommation électrique...

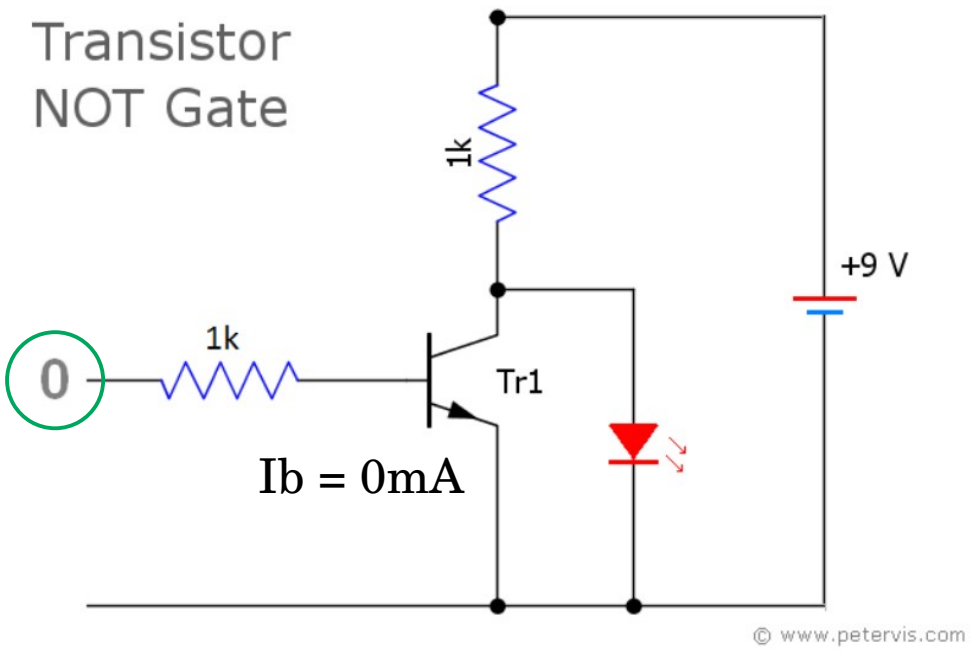
vitesse de calcul……..

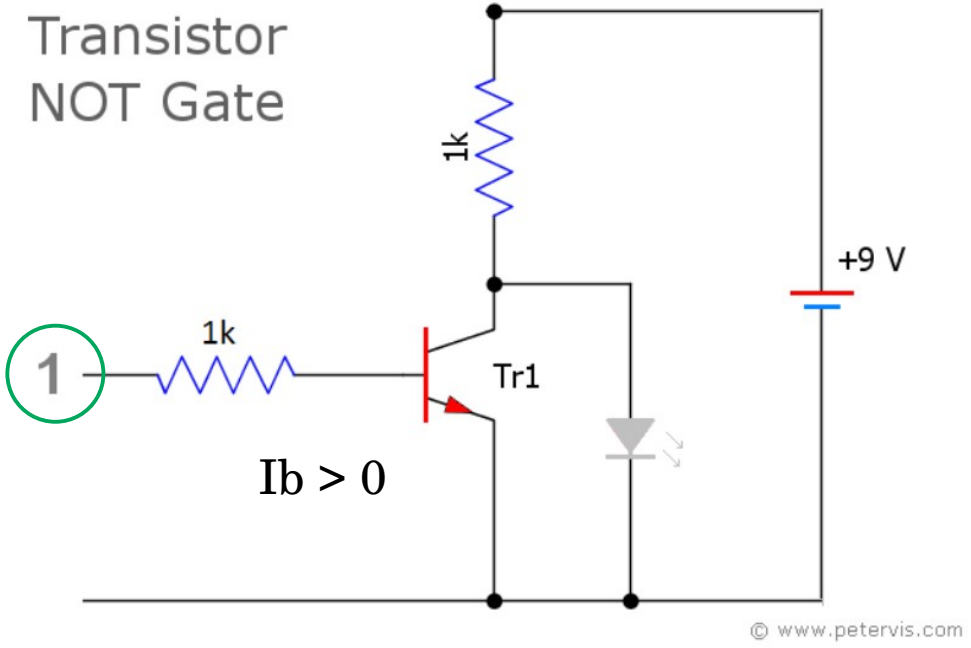
capacité de calcul…….

capacité de stockage……..

NOT : un inverseur (0 ⇒1, 1 ⇒ 0) avec un transistor

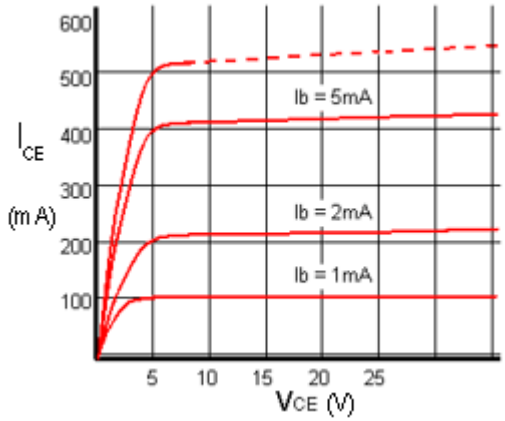
NOT : un inverseur (0 ⇒1, 1 ⇒ 0) avec un transistor



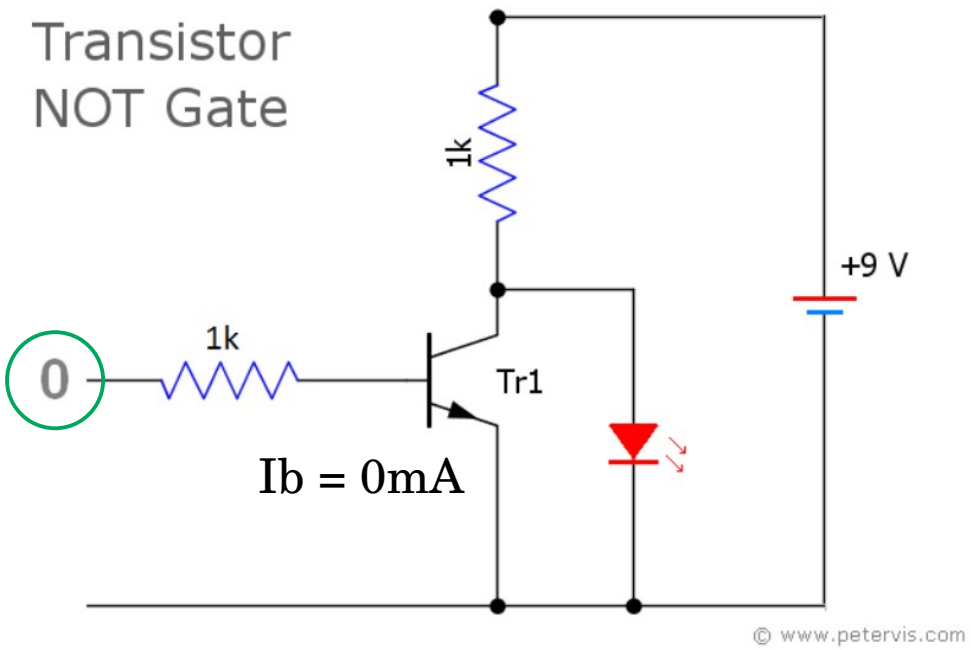


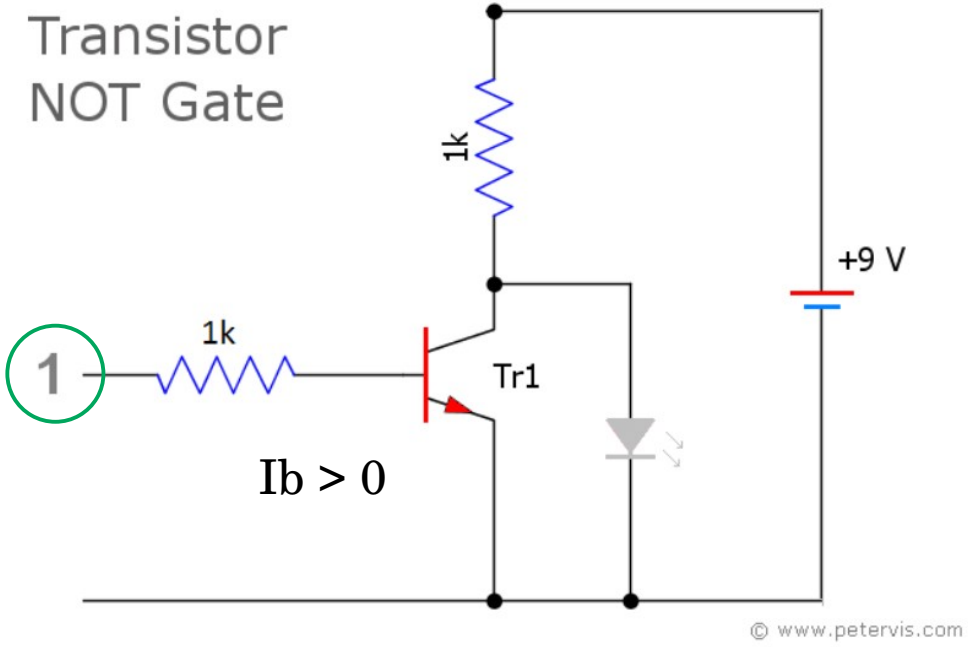
NOT

x	F
0	1
1	0



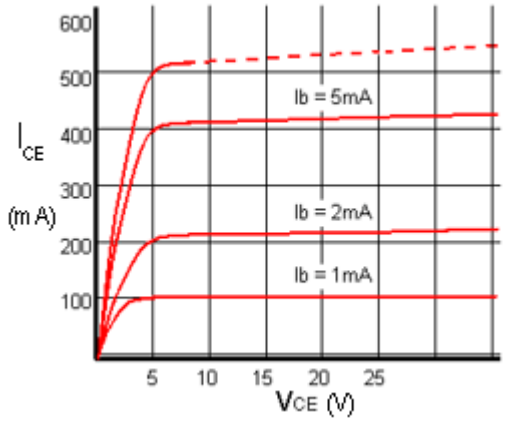
NOT : un inverseur (0 ⇒1, 1 ⇒ 0) avec un transistor





NOT

x	F
0	1
1	0



Un bit physique = un **commutateur électronique**.

Un qubit physique =

un **système quantique**.

Un qubit physique =

un **système quantique**.

Valeurs Booléenne :

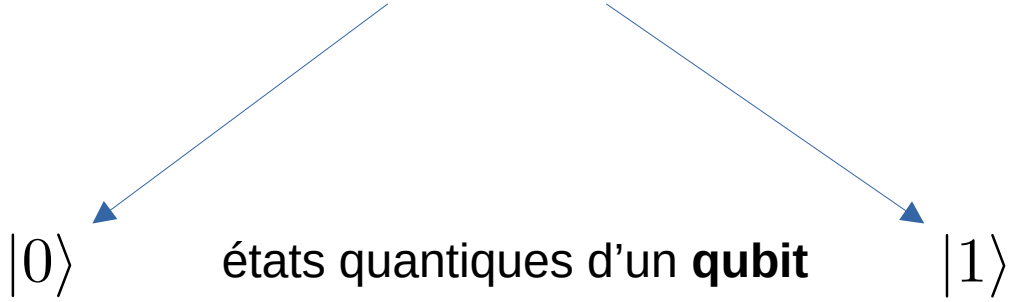
$\{0 \quad , \quad 1\}$

Un qubit physique =

un **système quantique**.

Valeurs Booléenne :

{0 , 1}



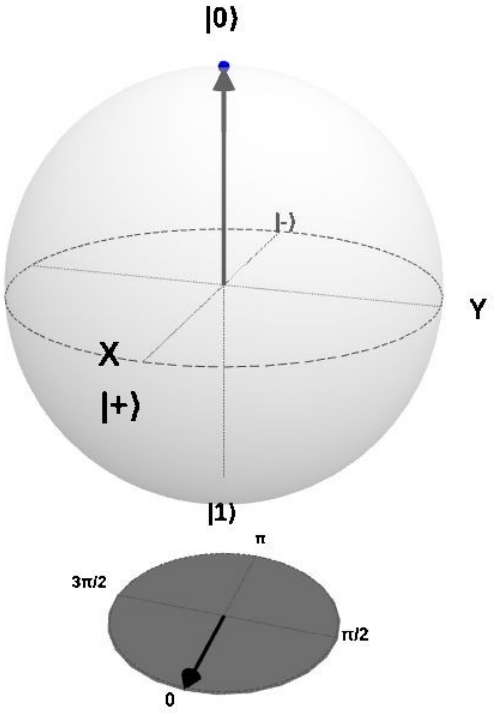
Un qubit physique =

un **système quantique**.

Valeurs Booléenne :

$\{0 \quad , \quad 1\}$

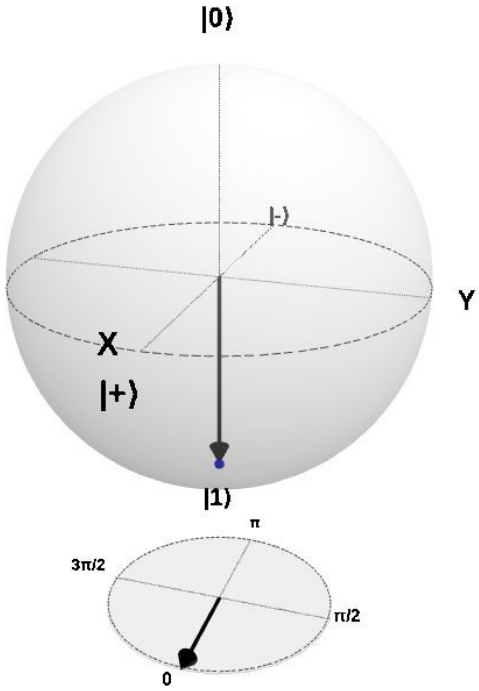
$|0\rangle$ états quantiques d'un **qubit** $|1\rangle$



représentation

sur la

Sphère de Bloch



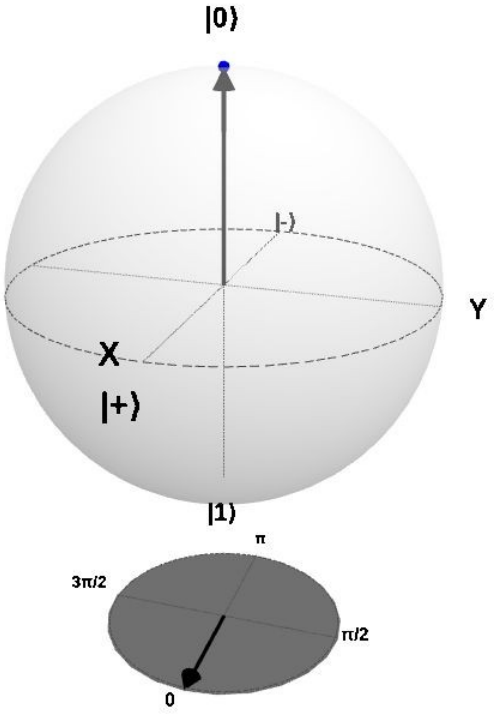
Un qubit physique =

un **système quantique**.

Valeurs Booléenne :

{0 , 1}

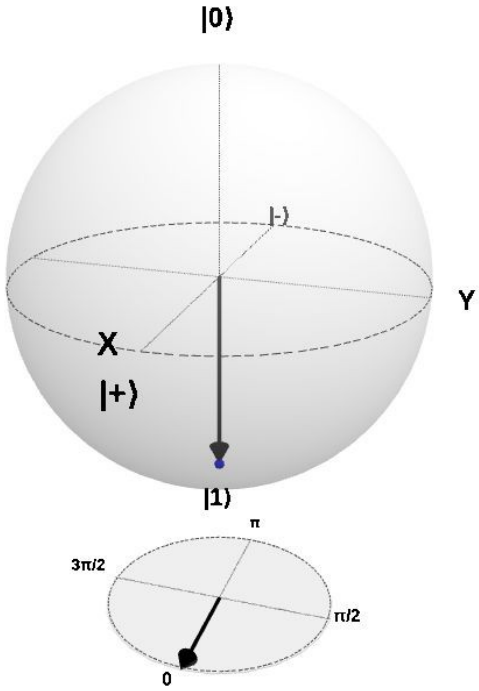
$|0\rangle$ états quantiques d'un **qubit** $|1\rangle$



représentation

sur la

Sphère de Bloch



L'équivalent du **NOT**

$$X|0\rangle = |1\rangle \quad , \quad X|1\rangle = |0\rangle$$

(« ket » dans la notation de Dirac)

Source: https://javafxpert.github.io/grok-bloch/

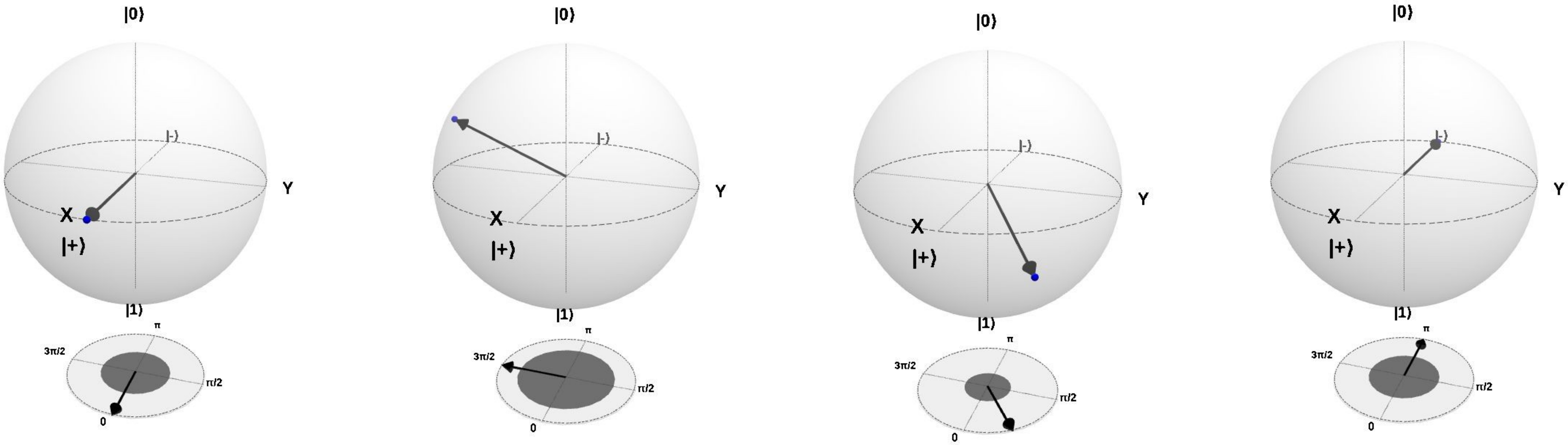
Opérations sur les qubits

Opérations sur les qubits

En général, des **rotations arbitraires** du vecteur représentant le qubit sur **toute la surface** de la Sphère de Bloch : une infinité de possibilités ?!

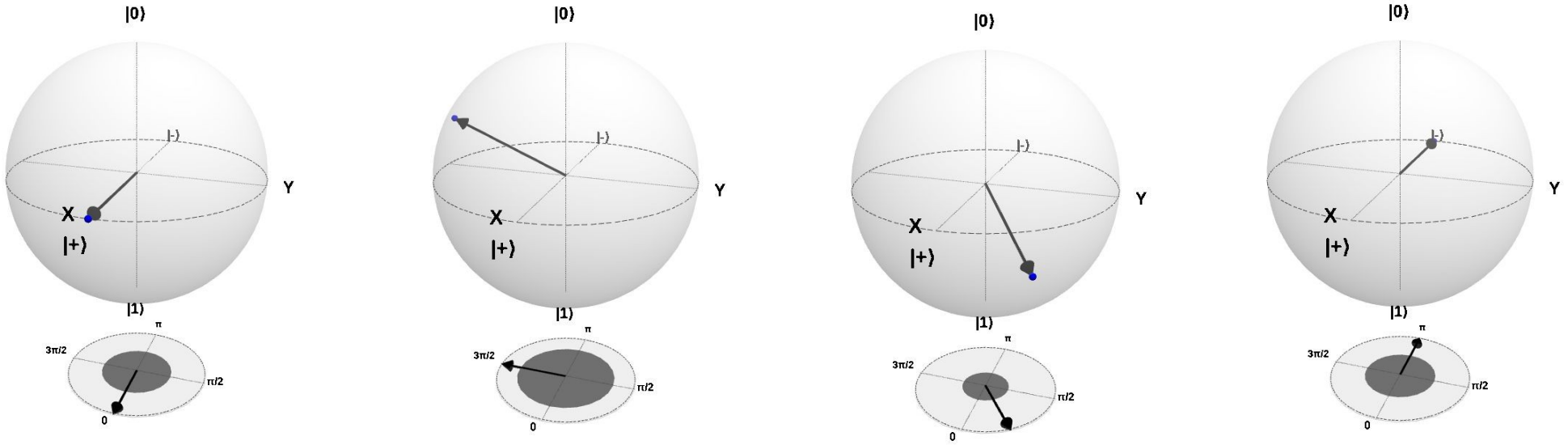
Opérations sur les qubits

En général, des **rotations arbitraires** du vecteur représentant le qubit sur **toute la surface** de la Sphère de Bloch : une infinité de possibilités ?!



Opérations sur les qubits

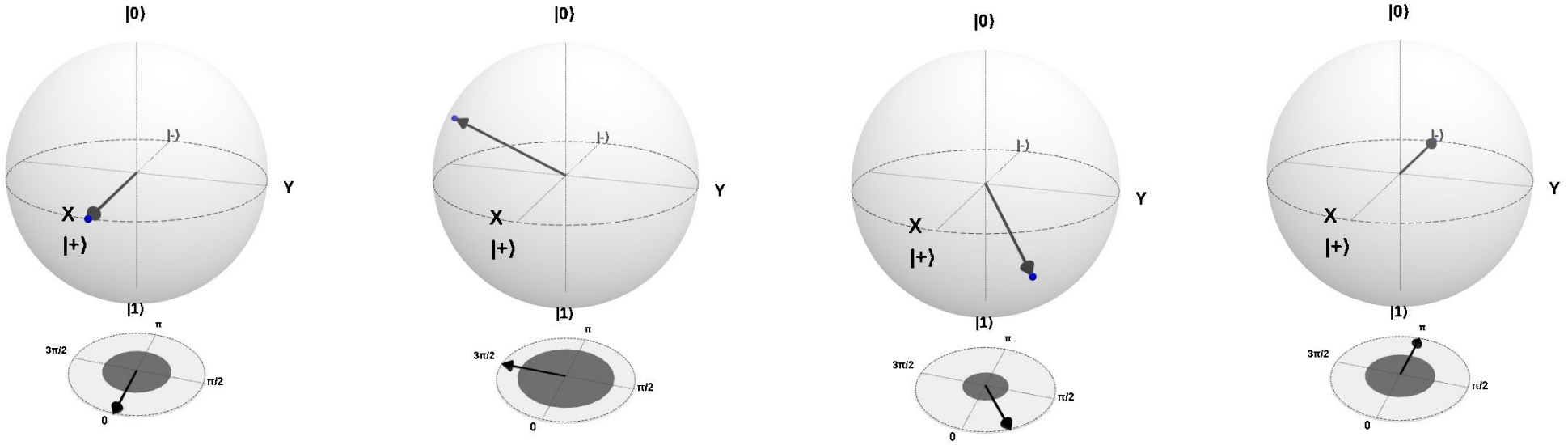
En général, des **rotations arbitraires** du vecteur représentant le qubit sur **toute la surface** de la Sphère de Bloch : une infinité de possibilités ?!



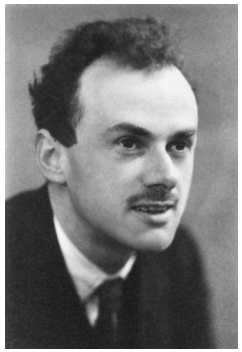
$$|0\rangle \rightarrow \cos\theta \;\; |0\rangle + e^{i\phi} \sin\theta \;\; |1\rangle \;\; , \;\; \{|0\rangle \; , \; |1\rangle\} = \text{ une base de vecteurs d'état}$$

Opérations sur les qubits

En général, des **rotations arbitraires** du vecteur représentant le qubit sur **toute la surface** de la Sphère de Bloch : une infinité de possibilités ?!



$$|0\rangle \rightarrow \cos\theta \;\; |0\rangle + e^{i\phi} \sin\theta \;\; |1\rangle \;\; , \;\; \{|0\rangle \; , \; |1\rangle\} = \text{ une base de vecteurs d'état}$$



P.A.M.Dirac
1902-1984

⇒ le principe de **superposition** : un système quantique peut se trouver à un moment donné simultanément dans plusieurs états.

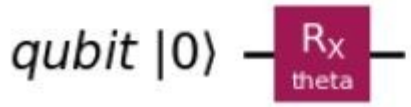
Question : qu'est-ce qu'on observe quand on fait une mesure ?

Exemple de code avec Qiskit

Exemple de code avec Qiskit

```
from qiskit import QuantumRegister, QuantumCircuit
from qiskit.circuit import Parameter
```

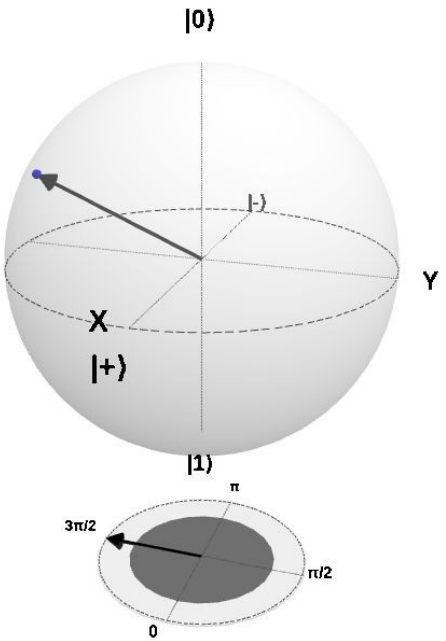
```
qr = QuantumRegister(1, 'qubit')
qc = QuantumCircuit(qr)
th_par = Parameter("theta")
qc.rx(th_par, [0])
qc.draw('mpl', initial_state=True)
```

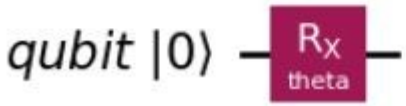


Exemple de code avec Qiskit

```
from qiskit import QuantumRegister, QuantumCircuit
from qiskit.circuit import Parameter
```

```
qr = QuantumRegister(1, 'qubit')
qc = QuantumCircuit(qr)
th_par = Parameter("theta")
qc.rx(th_par, [0])
qc.draw('mpl', initial_state=True)
```



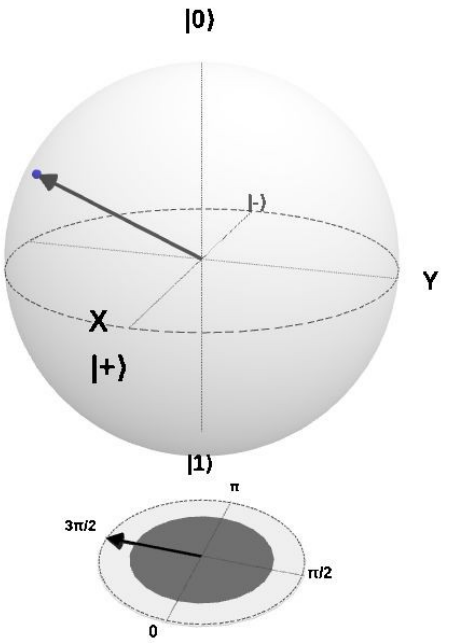


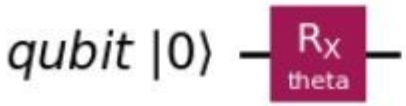
$$\Rightarrow \quad |\psi\rangle = \cos\theta \quad |0\rangle + \sin\theta \quad |1\rangle$$

Exemple de code avec Qiskit

```
from qiskit import QuantumRegister, QuantumCircuit
from qiskit.circuit import Parameter
```

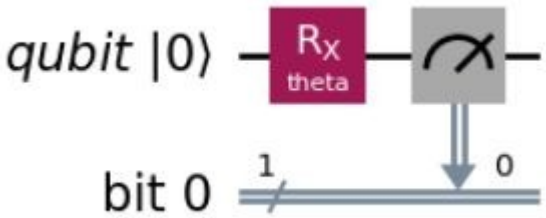
```
qr = QuantumRegister(1, 'qubit')
qc = QuantumCircuit(qr)
th_par = Parameter("theta")
qc.rx(th_par, [0])
qc.draw('mpl', initial_state=True)
```





$$\Rightarrow \quad |\psi\rangle = \cos\theta \;\; |0\rangle + \sin\theta \;\; |1\rangle$$

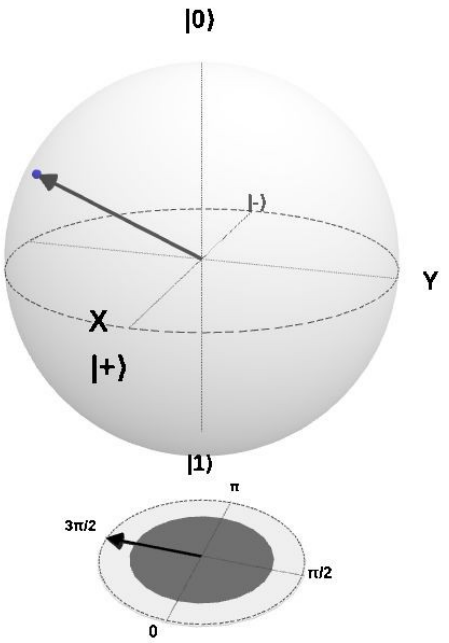
La **mesure** d'un état en **superposition** :

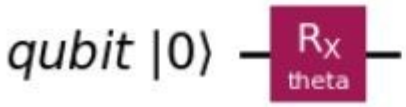


Exemple de code avec Qiskit

```
from qiskit import QuantumRegister, QuantumCircuit
from qiskit.circuit import Parameter
```

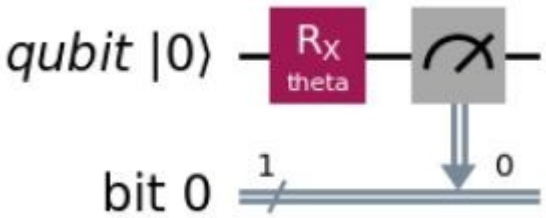
```
qr = QuantumRegister(1, 'qubit')
qc = QuantumCircuit(qr)
th_par = Parameter("theta")
qc.rx(th_par, [0])
qc.draw('mpl', initial_state=True)
```





$$\Rightarrow \quad |\psi\rangle = \cos\theta \;\; |0\rangle + \sin\theta \;\; |1\rangle$$

La **mesure** d'un état en **superposition** :



$$\Rightarrow \begin{cases} \mathrm{m} = 0 \;\; , \;\; |\psi\rangle_{m=0} = |0\rangle \;\; , \;\; \mathrm{prob} = (\cos\theta)^2 \\ \mathrm{m} = 1 \;\; , \;\; |\psi\rangle_{m=1} = |1\rangle \;\; , \;\; \mathrm{prob} = (\sin\theta)^2 \end{cases}$$

https://www.ibm.com/quantum/qiskit

Opérer avec plusieurs qubits

Opérer avec plusieurs qubits

L'opérateur **Hadamard** : $H\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle) \equiv \left|+\right\rangle, \quad H\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) \equiv \left|-\right\rangle$

Opérer avec plusieurs qubits

L'opérateur **Hadamard** : $H\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle) \equiv \left|+\right\rangle, \quad H\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) \equiv \left|-\right\rangle$

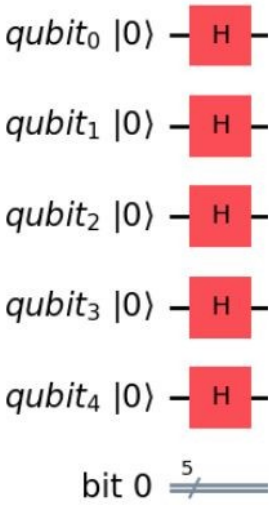
Un **registre** de qubits : $\left|q_4\right\rangle \otimes \left|q_3\right\rangle \otimes \left|q_2\right\rangle \otimes \left|q_1\right\rangle \otimes \left|q_0\right\rangle \equiv \left|q_4\ q_3\ \ q_2\ q_1\ q_0\right\rangle$

Opérer avec plusieurs qubits

L'opérateur **Hadamard** : $H\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle\,, \quad H\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle$

Un **registre** de qubits : $|q_4\rangle \otimes |q_3\rangle \otimes |q_2\rangle \otimes |q_1\rangle \otimes |q_0\rangle \equiv |q_4\ q_3\ \ q_2\ q_1\ q_0\rangle$

L'action du circuit (sans mesure) : $|0\ 0\ 0\ 0\ 0\rangle \rightarrow \frac{1}{\sqrt{2^5}} \sum_{q_4 \in \{0,1\}} \cdots \sum_{q_0 \in \{0,1\}} |q_4\ q_3\ \ q_2\ q_1\ q_0\rangle$



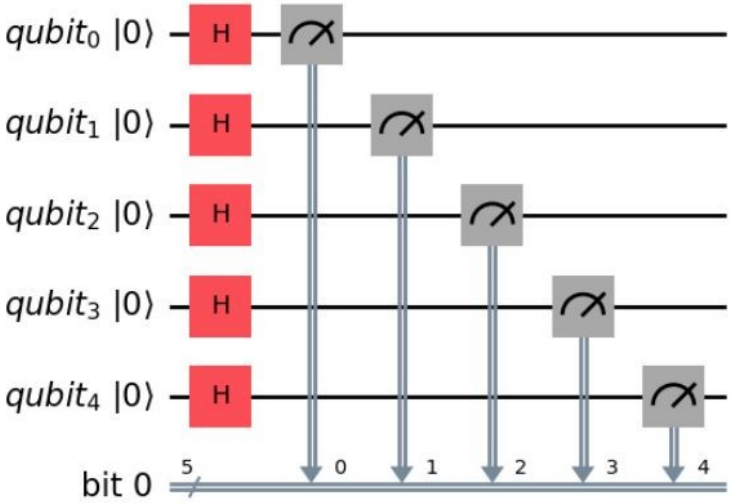
(il y a 32 termes dans cette somme)

Opérer avec plusieurs qubits

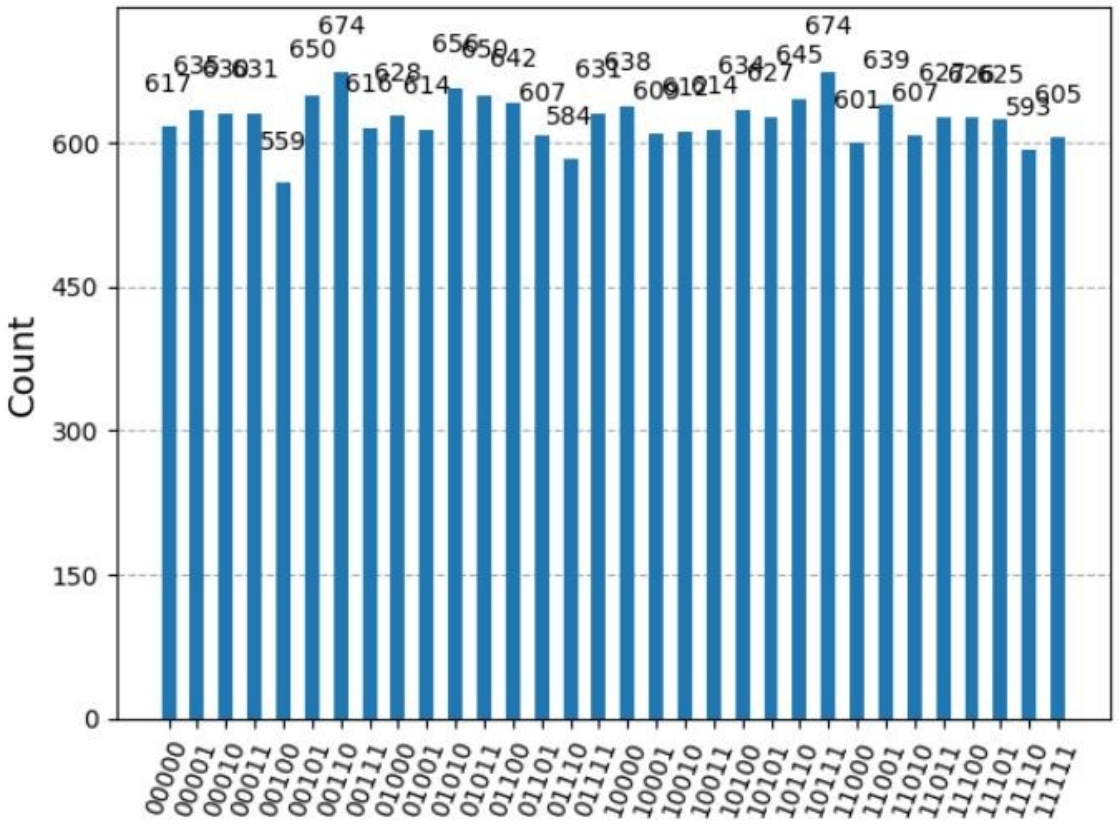
L'opérateur **Hadamard** : $H\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle) \equiv \left|+\right\rangle, \quad H\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) \equiv \left|-\right\rangle$

Un **registre** de qubits : $\left|q_4\right\rangle \otimes \left|q_3\right\rangle \otimes \left|q_2\right\rangle \otimes \left|q_1\right\rangle \otimes \left|q_0\right\rangle \equiv \left|q_4\ q_3\ \ q_2\ q_1\ q_0\right\rangle$

L'action du circuit (sans mesure) : $\left|0\ 0\ 0\ 0\ 0\right\rangle \rightarrow \frac{1}{\sqrt{2^5}} \sum_{q_4 \in \{0,1\}} \cdots \sum_{q_0 \in \{0,1\}} \left|q_4\ q_3\ \ q_2\ q_1\ q_0\right\rangle$



Les résultats de **20000 mesures** ⇒

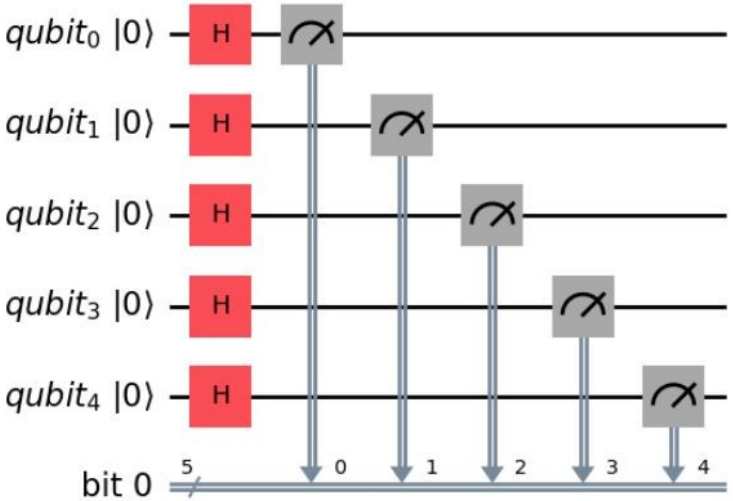


Opérer avec plusieurs qubits

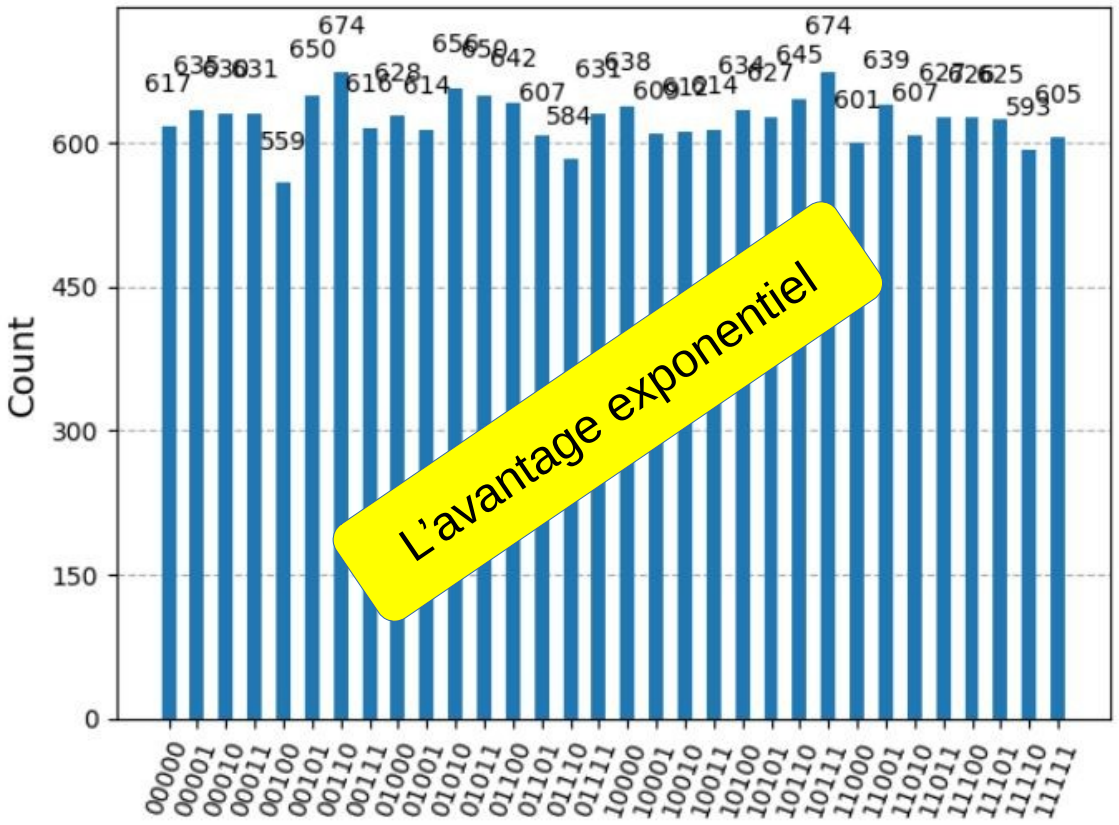
L'opérateur **Hadamard** : $H\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle\,, \quad H\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle$

Un **registre** de qubits : $|q_4\rangle \otimes |q_3\rangle \otimes |q_2\rangle \otimes |q_1\rangle \otimes |q_0\rangle \equiv |q_4\ q_3\ \ q_2\ q_1\ q_0\rangle$

L'action du circuit (sans mesure) : $|0\ 0\ 0\ 0\ 0\rangle \rightarrow \frac{1}{\sqrt{2^5}} \sum_{q_4 \in \{0,1\}} \cdots \sum_{q_0 \in \{0,1\}} |q_4\ q_3\ \ q_2\ q_1\ q_0\rangle$

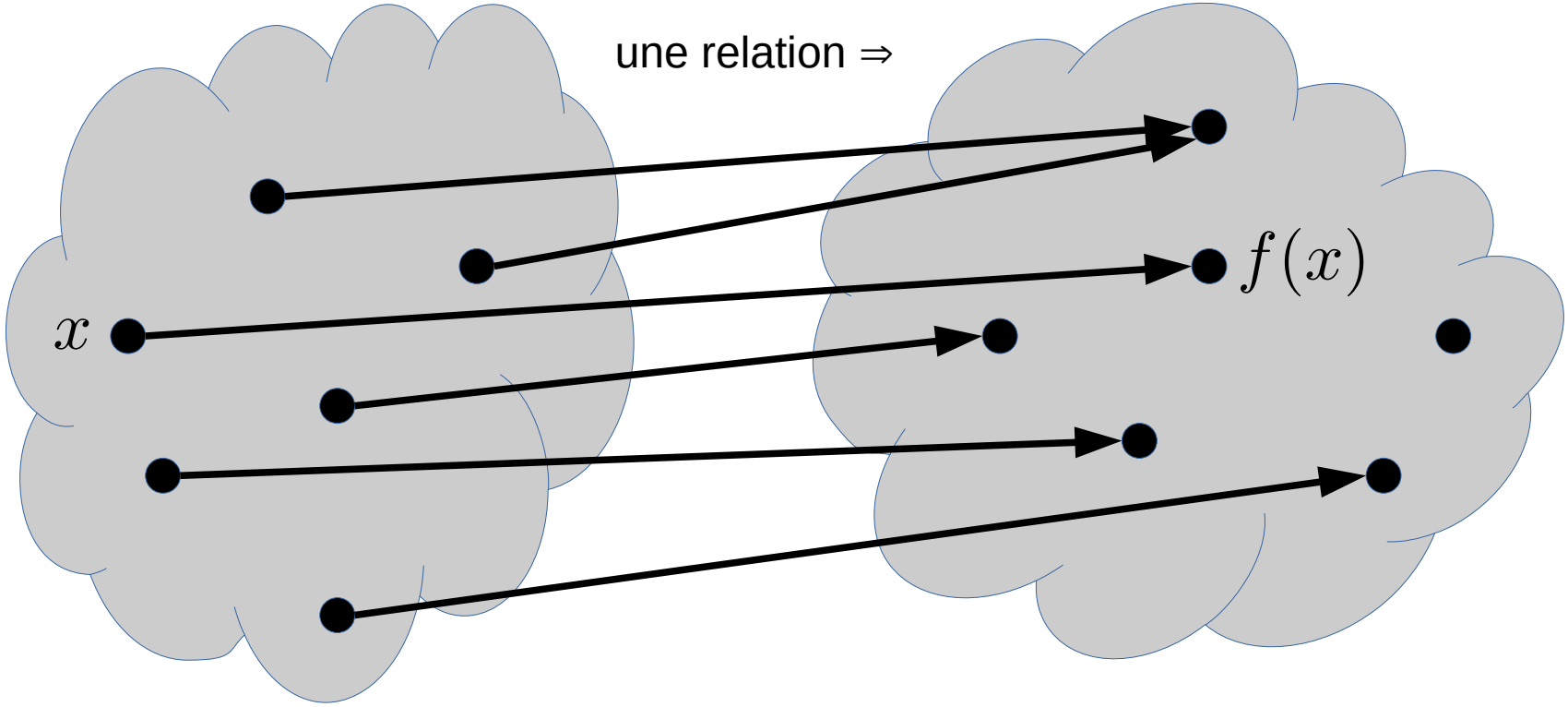


Les résultats de **20000 mesures** ⇒



Caractériser les fonctions ?

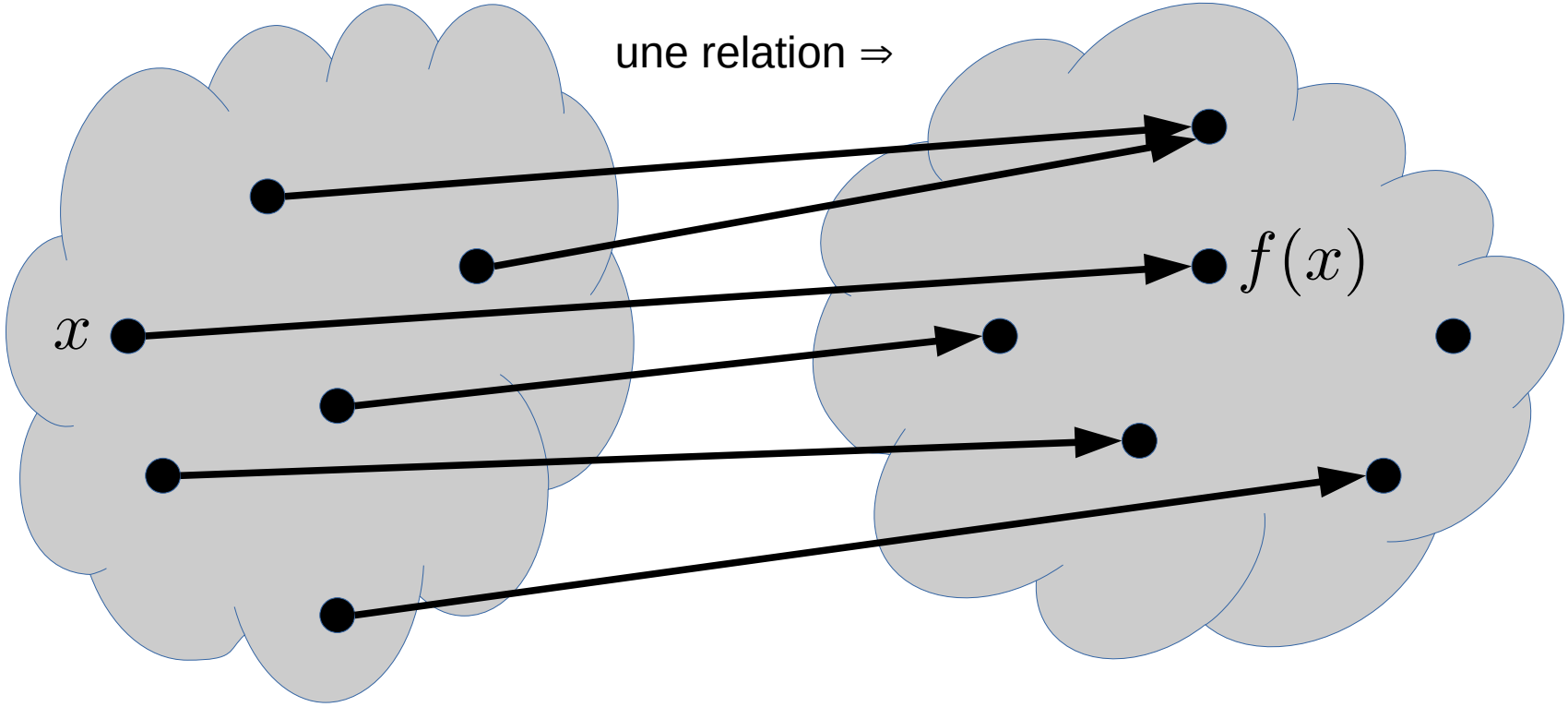
Caractériser les fonctions ?



Ensemble 1 : domaine de définition

Ensemble 2 : domaine de valeurs

Caractériser les fonctions ?



Ensemble 1 : domaine de définition

Ensemble 2 : domaine de valeurs

Exemple : $f : \{1, 2, 3, 4, 5, 6\} \to \{\text{rouge}, \text{bleu}, \text{noir}, \text{jaune}, \text{vert}, \text{blanc}\}$

$$f(1) = \text{blanc}, f(2) = \text{vert}, f(3) = \text{jaune}, f(4) = \text{bleu}, f(5) = \text{rouge}, f(6) = \text{noir}$$

L’algorithme Deutsch-Jozsa (1992)

L’algorithme Deutsch-Jozsa (1992)

Prenons la fonction $f : \{1, 2, 3, 4, 5, 6\} \to \{0, 1\}$ et considérons deux cas :

L’algorithme Deutsch-Jozsa (1992)

Prenons la fonction $f : \{1, 2, 3, 4, 5, 6\} \rightarrow \{0, 1\}$ et considérons deux cas :

a) une fonction **constante**

$$\begin{cases} \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 0 \\ \\ \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 1 \end{cases}$$

L’algorithme Deutsch-Jozsa (1992)

Prenons la fonction $f : \{1,2,3,4,5,6\} \to \{0,1\}$ et considérons deux cas :

a) une fonction **constante**

$$\begin{cases} \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 0 \\ \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 1 \end{cases}$$

b) une fonction **équilibrée**

$$\begin{cases} \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = 0 \ , \ \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 1 \\ \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = 1 \ , \ \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 0 \\ \mathrm{f}(2) = \mathrm{f}(4) = \mathrm{f}(6) = 0 \ , \ \mathrm{f}(1) = \mathrm{f}(3) = \mathrm{f}(5) = 1 \\ \mathrm{etc} \ \ldots \end{cases}$$

L’algorithme Deutsch-Jozsa (1992)

Prenons la fonction $f : \{1,2,3,4,5,6\} \to \{0,1\}$ et considérons deux cas :

a) une fonction **constante**

$$\begin{cases} \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 0 \\ \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 1 \end{cases}$$

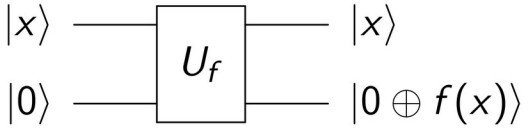
b) une fonction **équilibrée**

$$\begin{cases} \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = 0 \ , \ \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 1 \\ \mathrm{f}(1) = \mathrm{f}(2) = \mathrm{f}(3) = 1 \ , \ \mathrm{f}(4) = \mathrm{f}(5) = \mathrm{f}(6) = 0 \\ \mathrm{f}(2) = \mathrm{f}(4) = \mathrm{f}(6) = 0 \ , \ \mathrm{f}(1) = \mathrm{f}(3) = \mathrm{f}(5) = 1 \\ \mathrm{etc} \ \ldots \end{cases}$$

Combien d’évaluations de la fonction sont nécessaires pour déterminer si « a » ou « b » ?

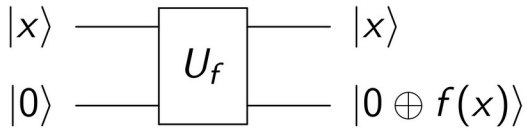

David Deutsch et Richard Jozsa

un « oracle » de fonction

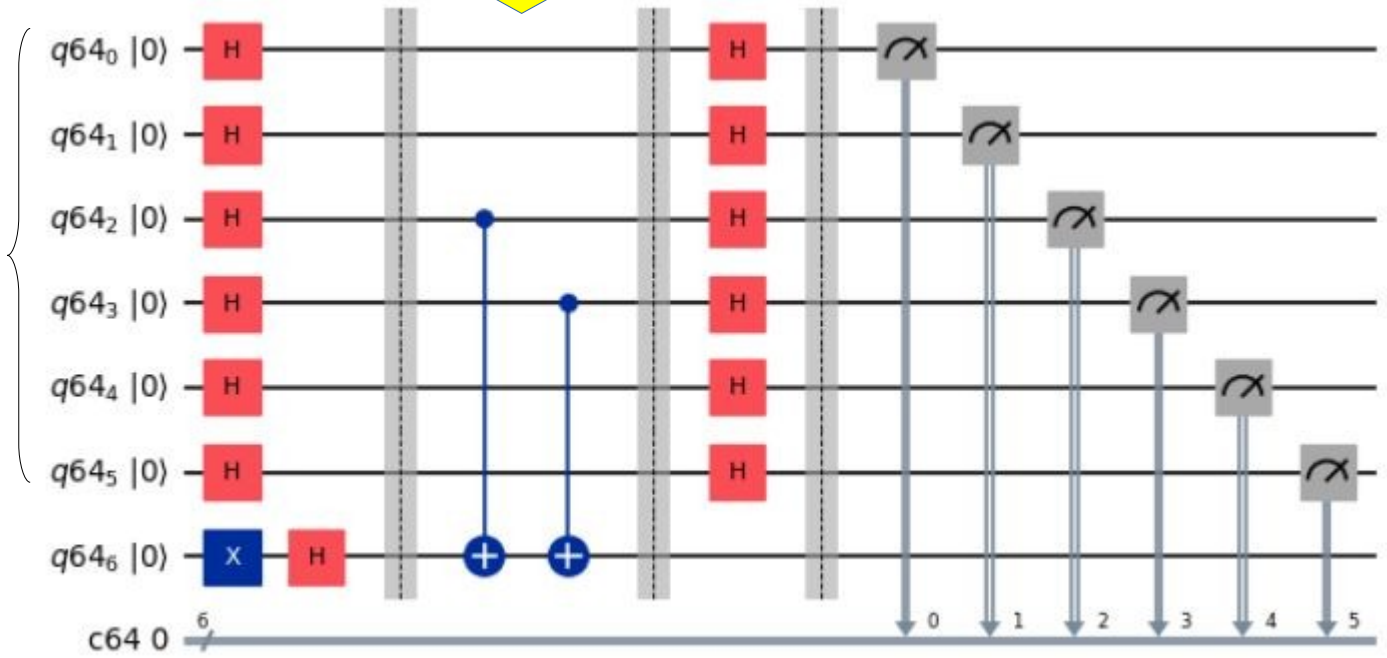



David Deutsch et Richard Jozsa

un « oracle » de fonction

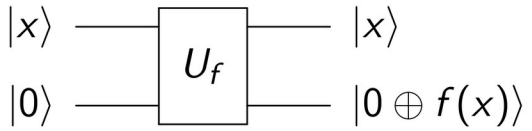


Un qubit auxiliaire pour réaliser l'**oracle** d'évaluation de la fonction ⇒

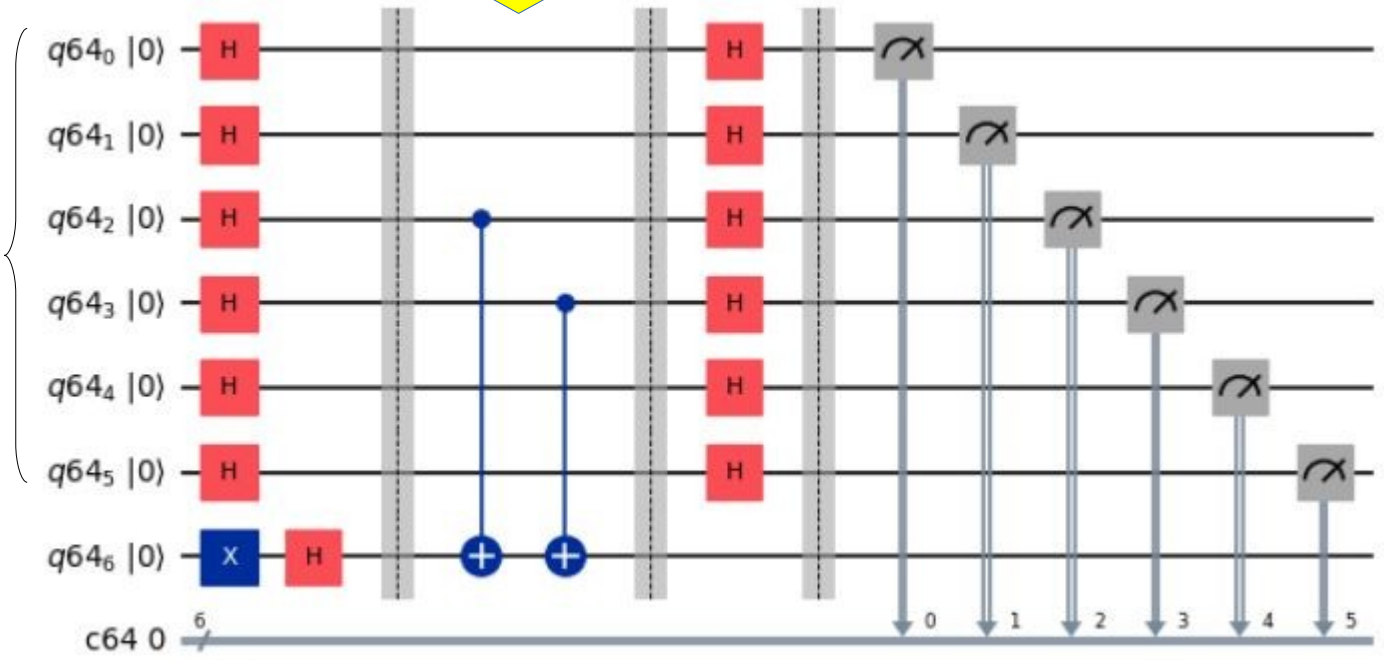



David Deutsch et Richard Jozsa

un « oracle » de fonction



Un qubit auxiliaire pour réaliser l'**oracle** d'évaluation de la fonction ⇒

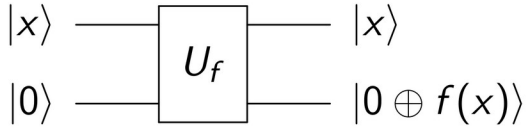


Exemple d'une fonction (ici équilibrée) avec 6 arguments, donc $2^6 = 64$ valeurs possible :

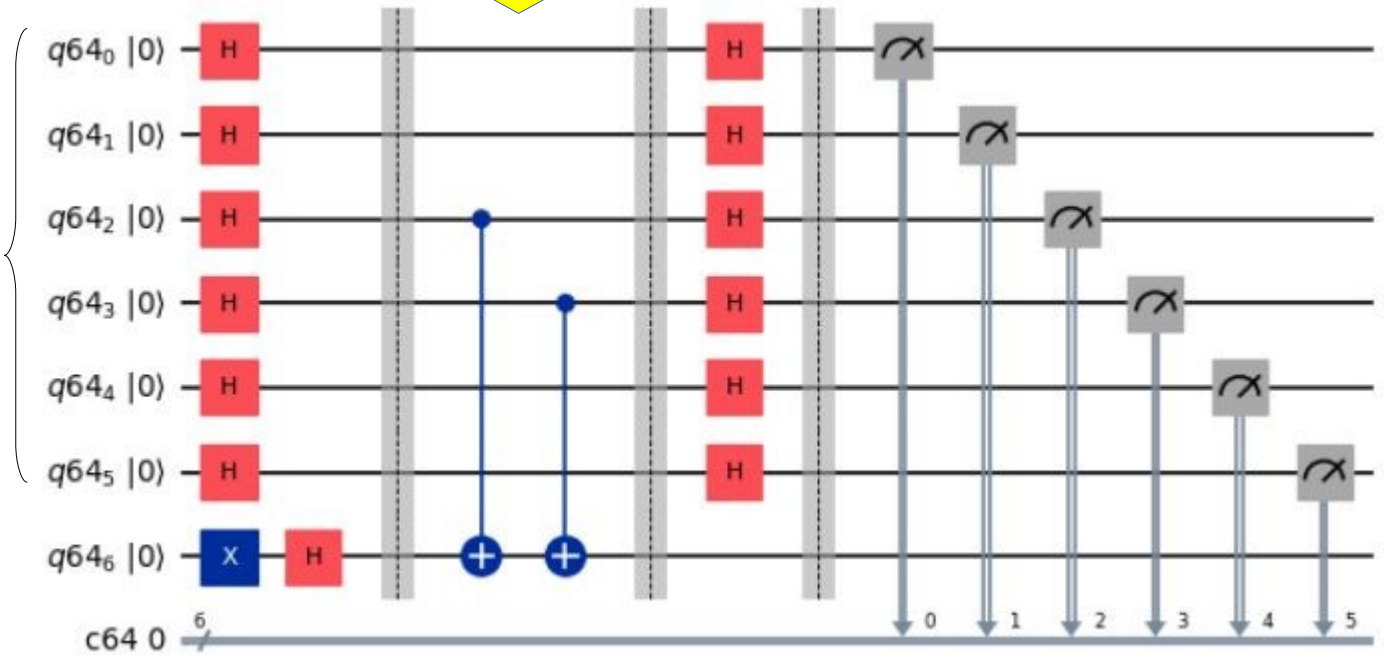
$$f : \{0, 1, 2, \ldots, 63\} \to \{0, 1\}$$


David Deutsch et Richard Jozsa

un « oracle » de fonction



Un qubit auxiliaire pour réaliser l'**oracle** d'évaluation de la fonction ⇒



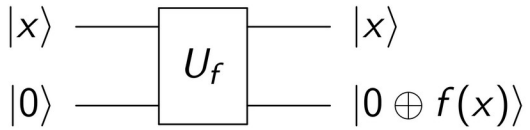
Exemple d'une fonction (ici équilibrée) avec 6 arguments, donc $2^6 = 64$ valeurs possible :

$$f : \{0, 1, 2, \ldots, 63\} \rightarrow \{0, 1\}$$

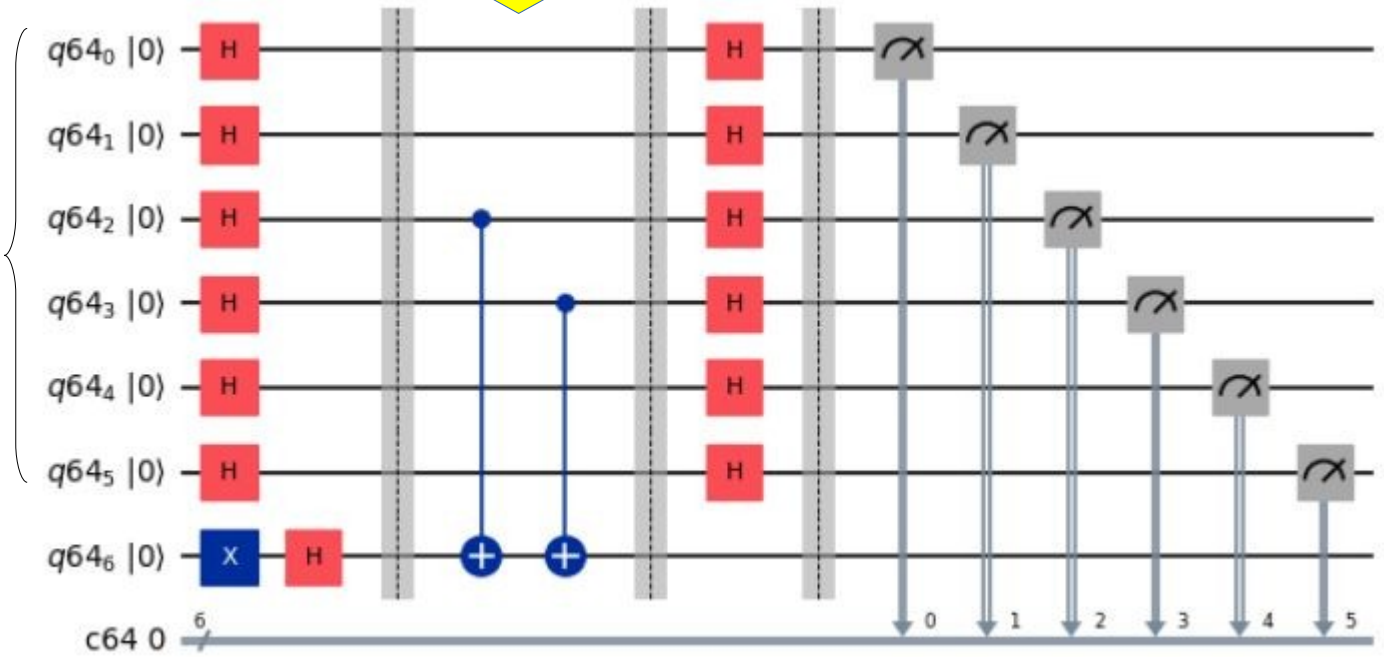
une seule mesure est nécessaire : si le résultat de la mesure **n'est pas** '000000', alors la fonction est équilibrée. Dans le cas contraire, la fonction est constante.


David Deutsch et Richard Jozsa

un « oracle » de fonction



Un qubit auxiliaire pour réaliser l'**oracle** d'évaluation de la fonction ⇒



Exemple d'une fonction (ici équilibrée) avec 6 arguments, donc $2^6 = 64$ valeurs possible :

$$f : \{0, 1, 2, \ldots, 63\} \rightarrow \{0, 1\}$$

une seule mesure est nécessaire : si le résultat de la mesure **n'est pas** '000000', alors la fonction est équilibrée. Dans le cas contraire, la fonction est constante.

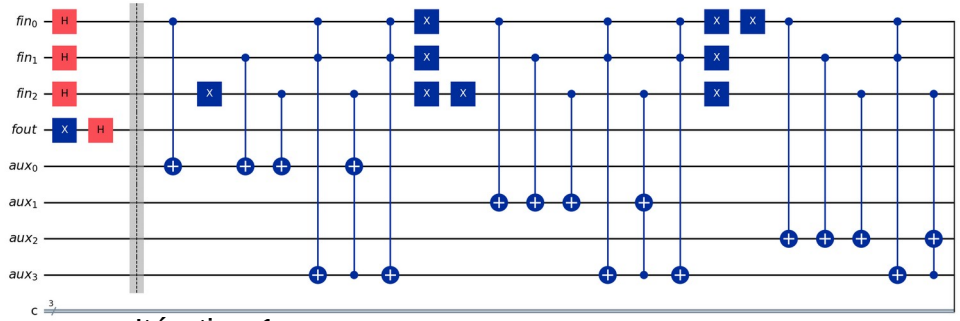
Nous obtenons une **accélération polynomiale** : $O(N) \rightarrow 1$

$O(N)$ pour dire "de l'ordre de" N

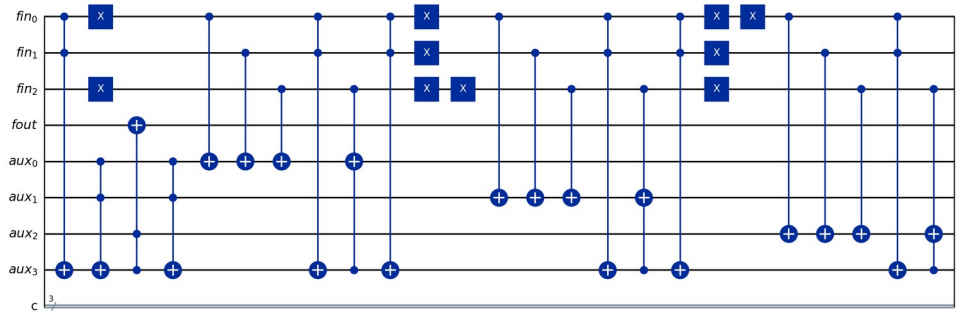
Sources : The Progress Network, Wikipedia

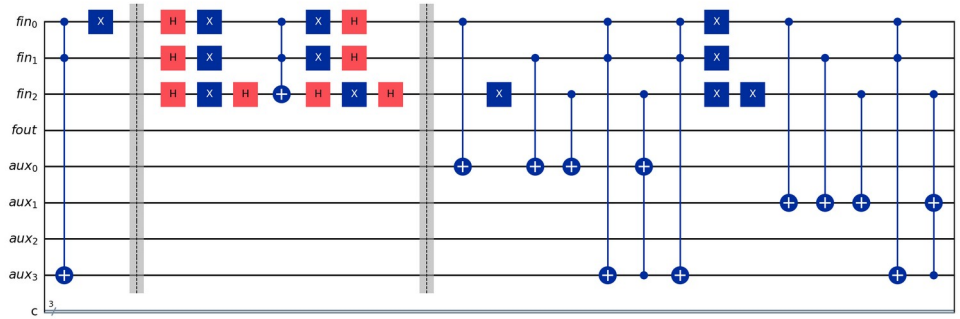
Comment chercher un élément dans une liste non triée ? L’algorithme de Grover (1996)

Comment chercher un élément dans une liste non triée ? L'algorithme de Grover (1996)



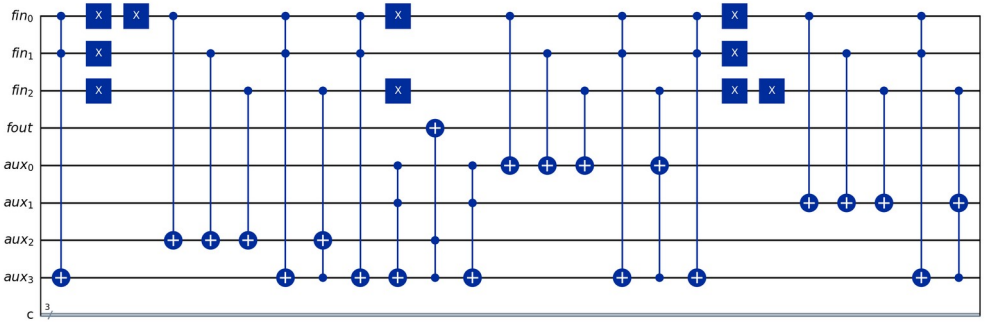
Itération 1

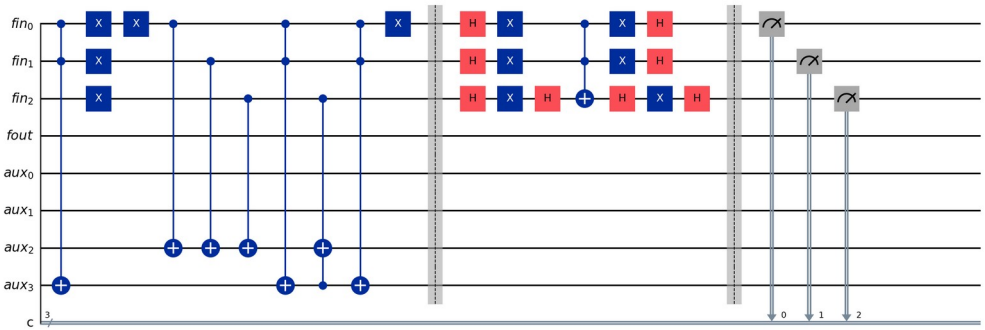




Itération 2

(cont.)





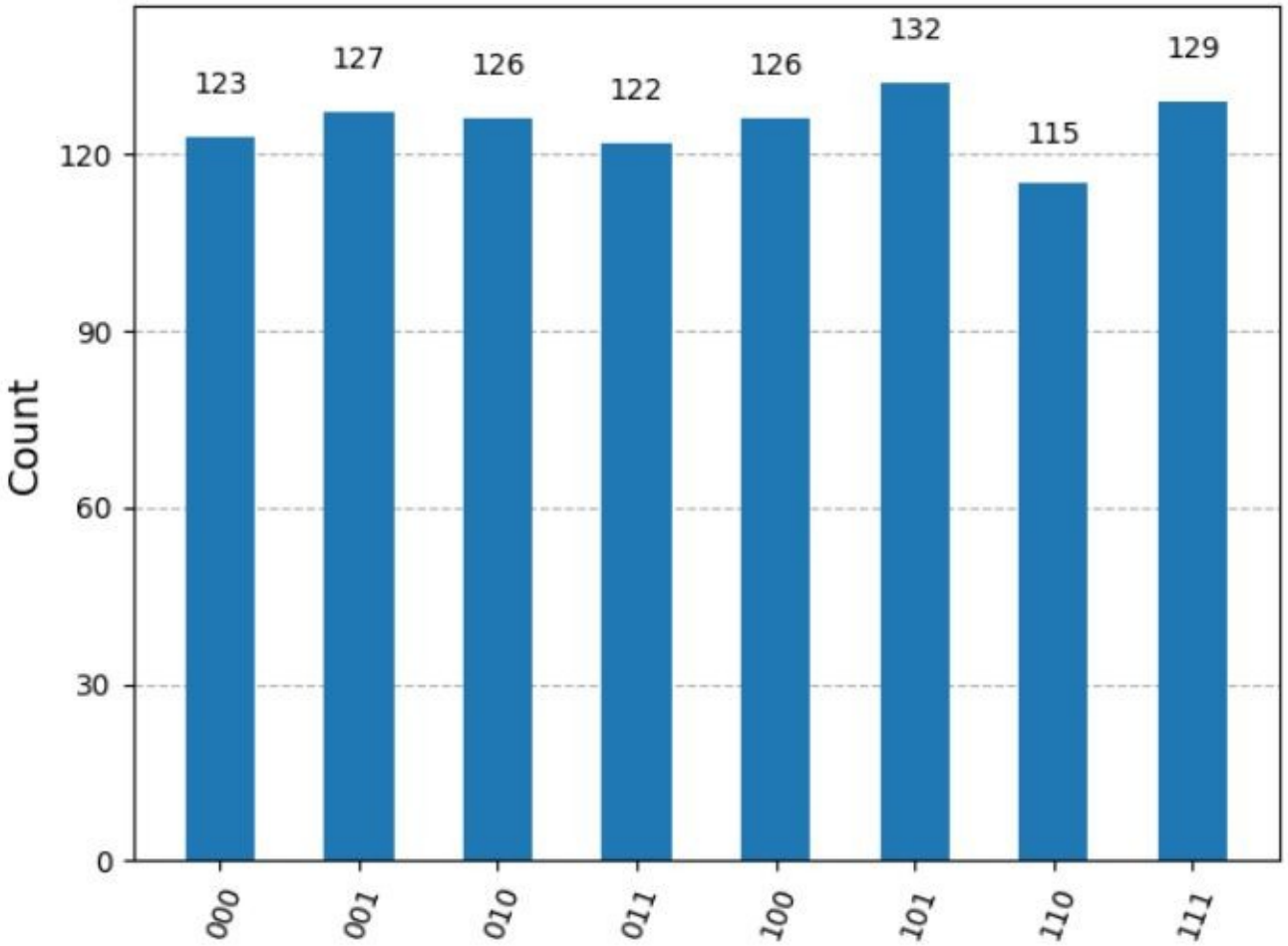
Sources : QUANTUMPEDIA

Recherche non-structurée dans une liste de 8 éléments de l'argument qui **satisfait** la fonction :

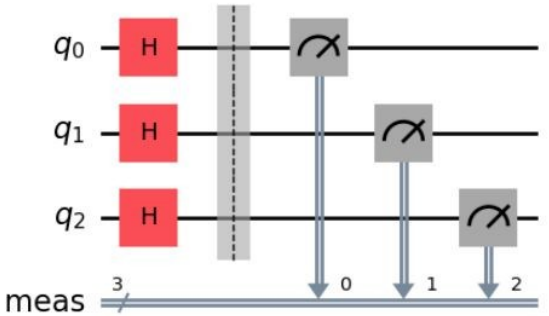
$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

Recherche non-structurée dans une liste de 8 éléments de l'argument qui **satisfait** la fonction :

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$



Première partie du circuit :

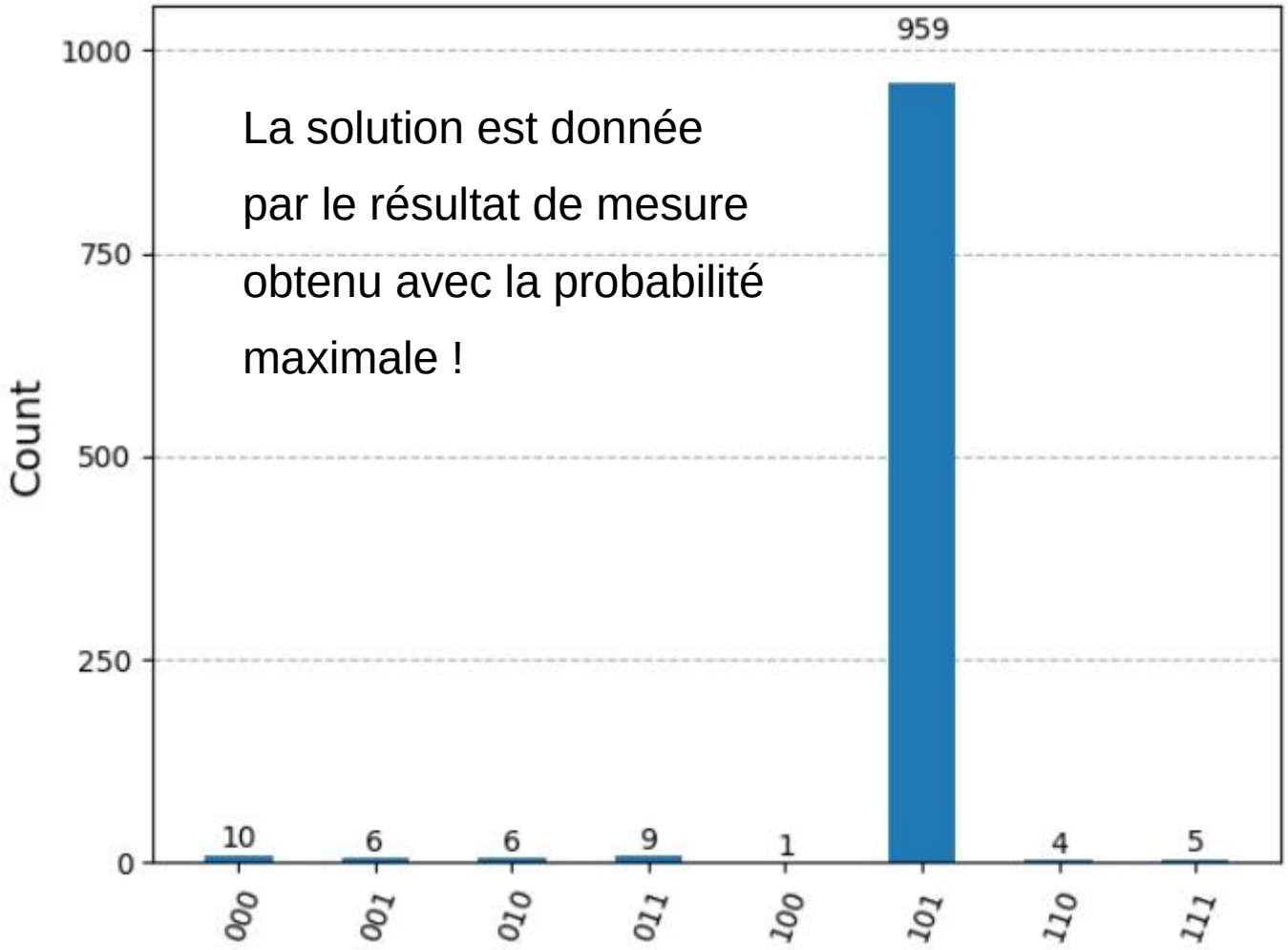


Initialement, toutes les valeurs possibles des arguments de la fonction étudiée sont mises en **superposition**, avec probabilité égale d'apparition dans une mesure.

$$f(1, 0, 1) = 1$$

Recherche non-structurée dans une liste de 8 éléments de l'argument qui **satisfait** la fonction :

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

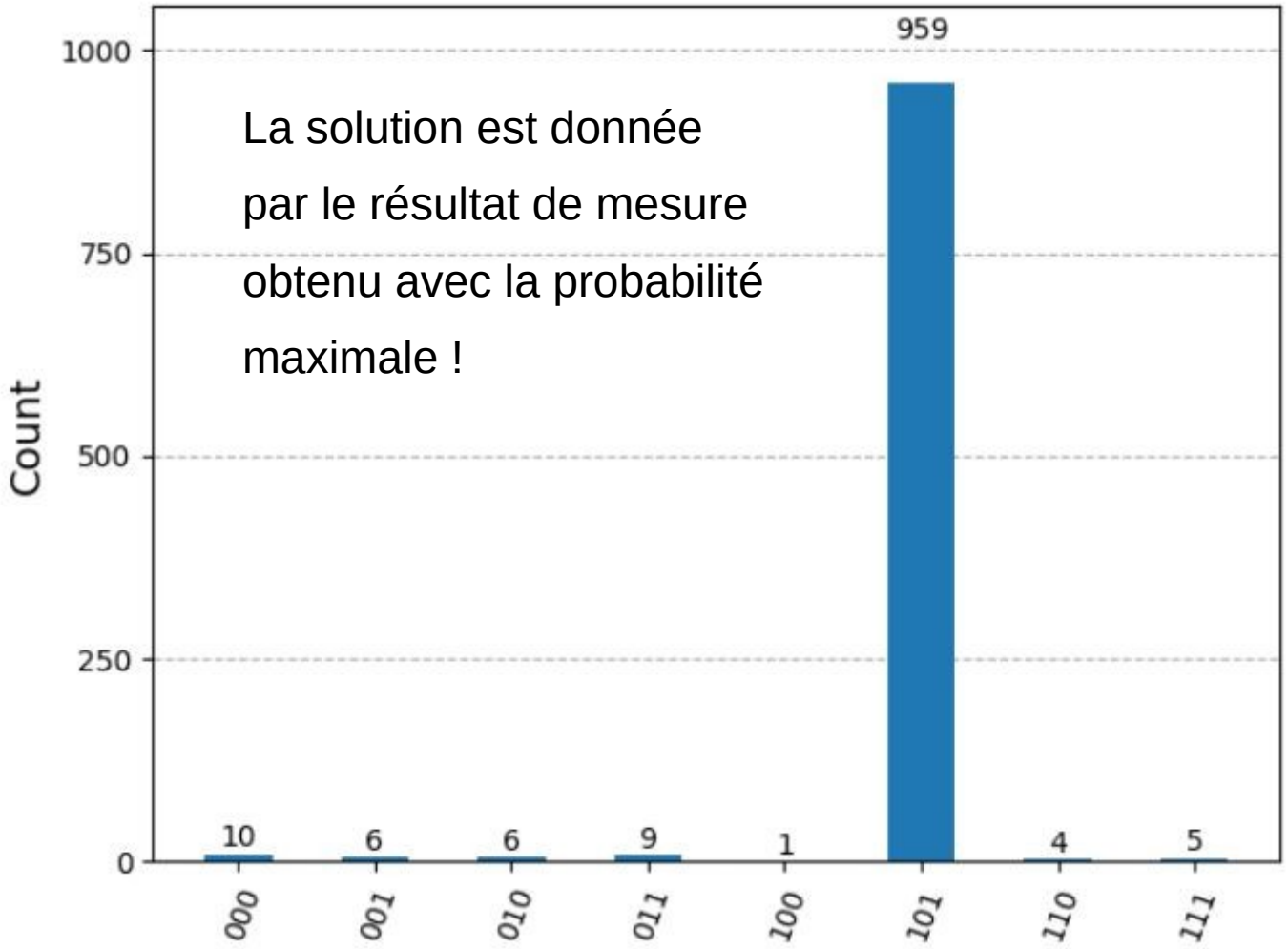


$f(1,0,1) = 1$

Après l'application de deux itérations du partie du circuit dite d'**amplification**, la solution ressort (clairement) le plus souvent parmi les autres possibilités.

Recherche non-structurée dans une liste de 8 éléments de l'argument qui **satisfait** la fonction :

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$



La solution est donnée par le résultat de mesure obtenu avec la probabilité maximale !

$f(1,0,1) = 1$

Avec n qubits on on représente $N = 2^n$ valeurs et le nombre d'itérations optimal est

$$k = \frac{\pi}{4}\sqrt{N}$$

Nous obtenons une **accélération quadratique**

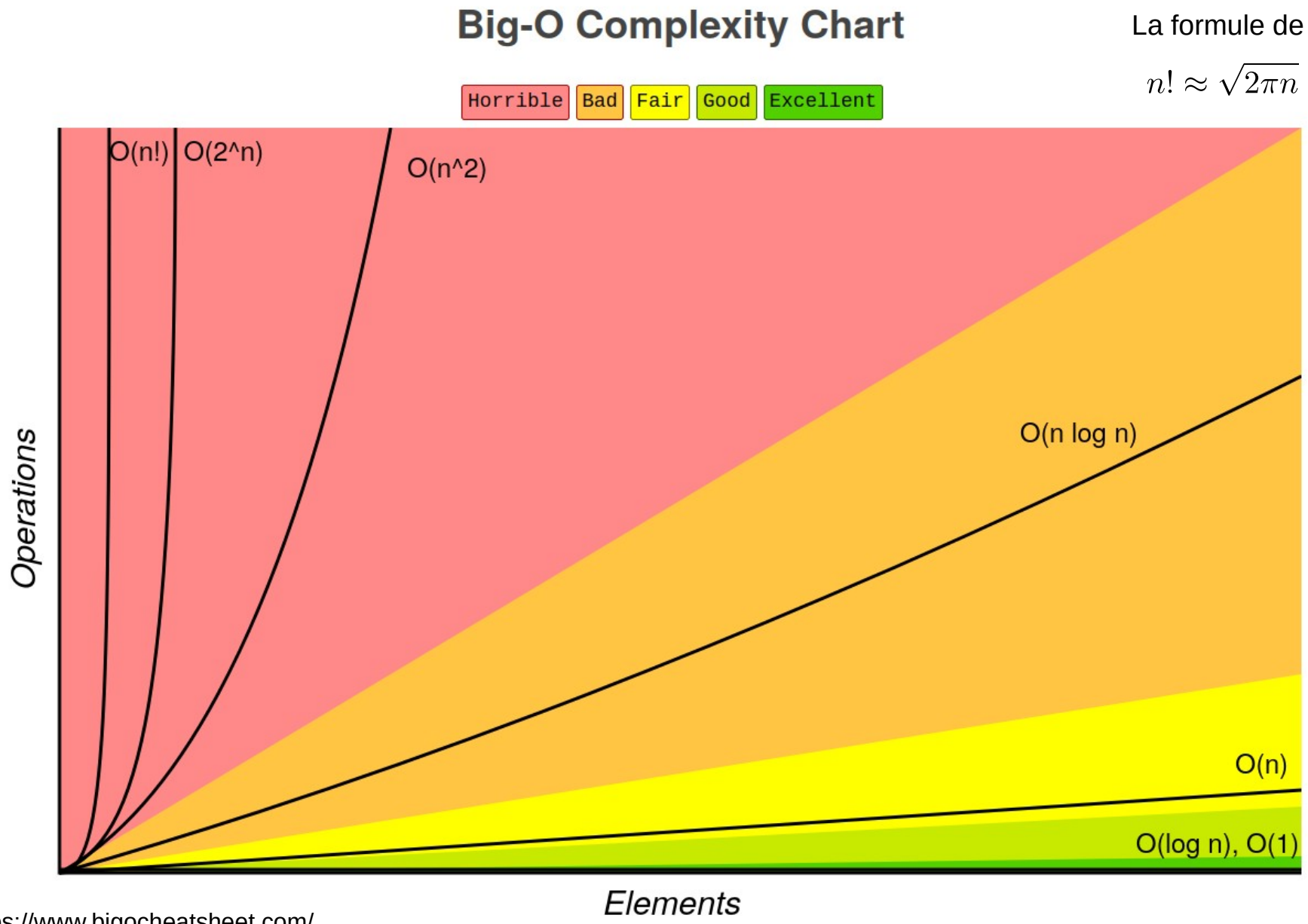
$$O(N) \rightarrow O(\sqrt{N})$$

le temps de calcul augmente selon $\sqrt{N}$ au lieu de N

À retenir : la complexité des calculs limite (surtout en temps de calcul) la dimension des problèmes qu'on peut (exactement) résoudre, même avec de grandes capacités de calcul.

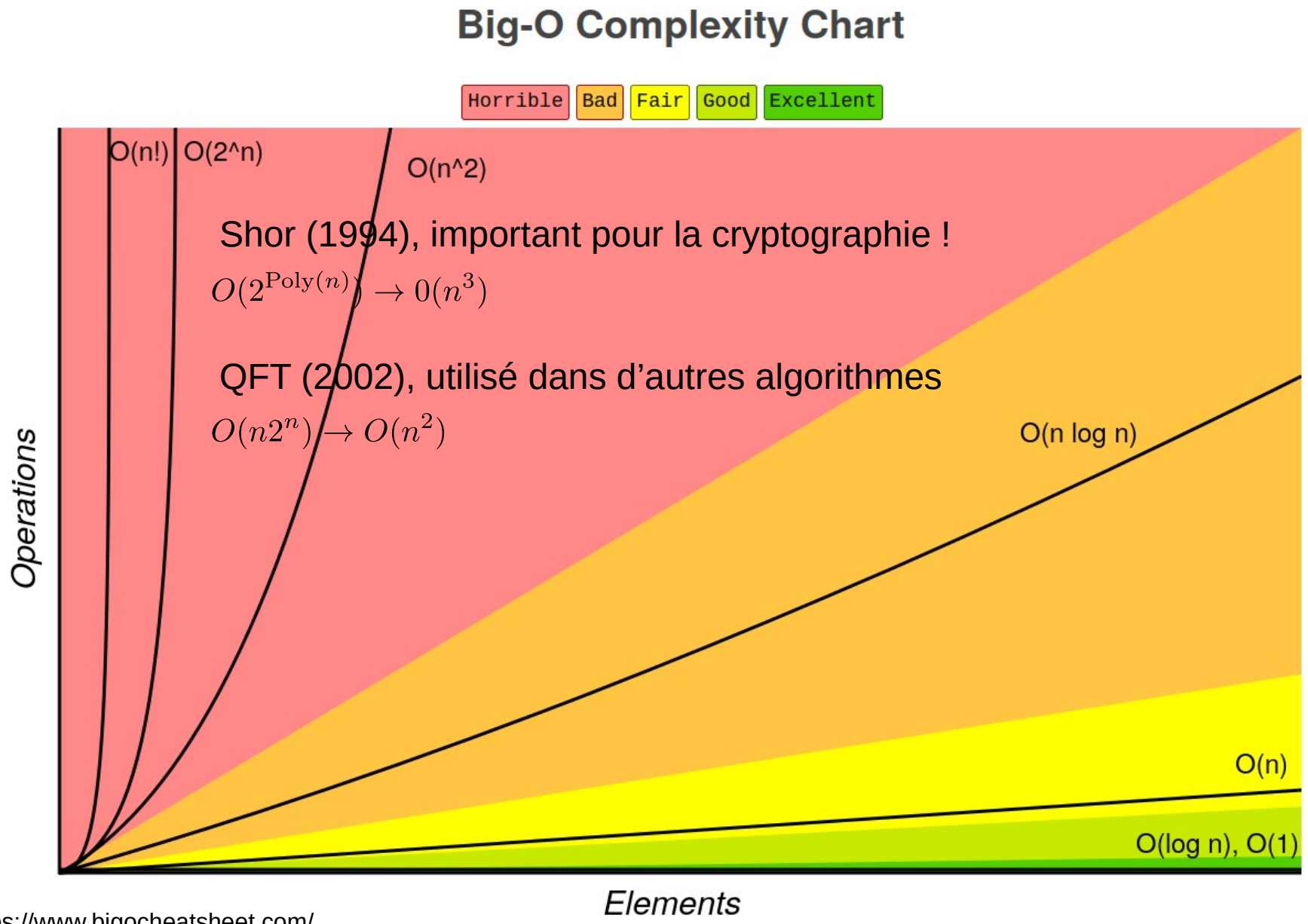
La formule de Stirling

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$



Source https://www.bigocheatsheet.com/

Autres algorithmes



Shor (1994), important pour la cryptographie !

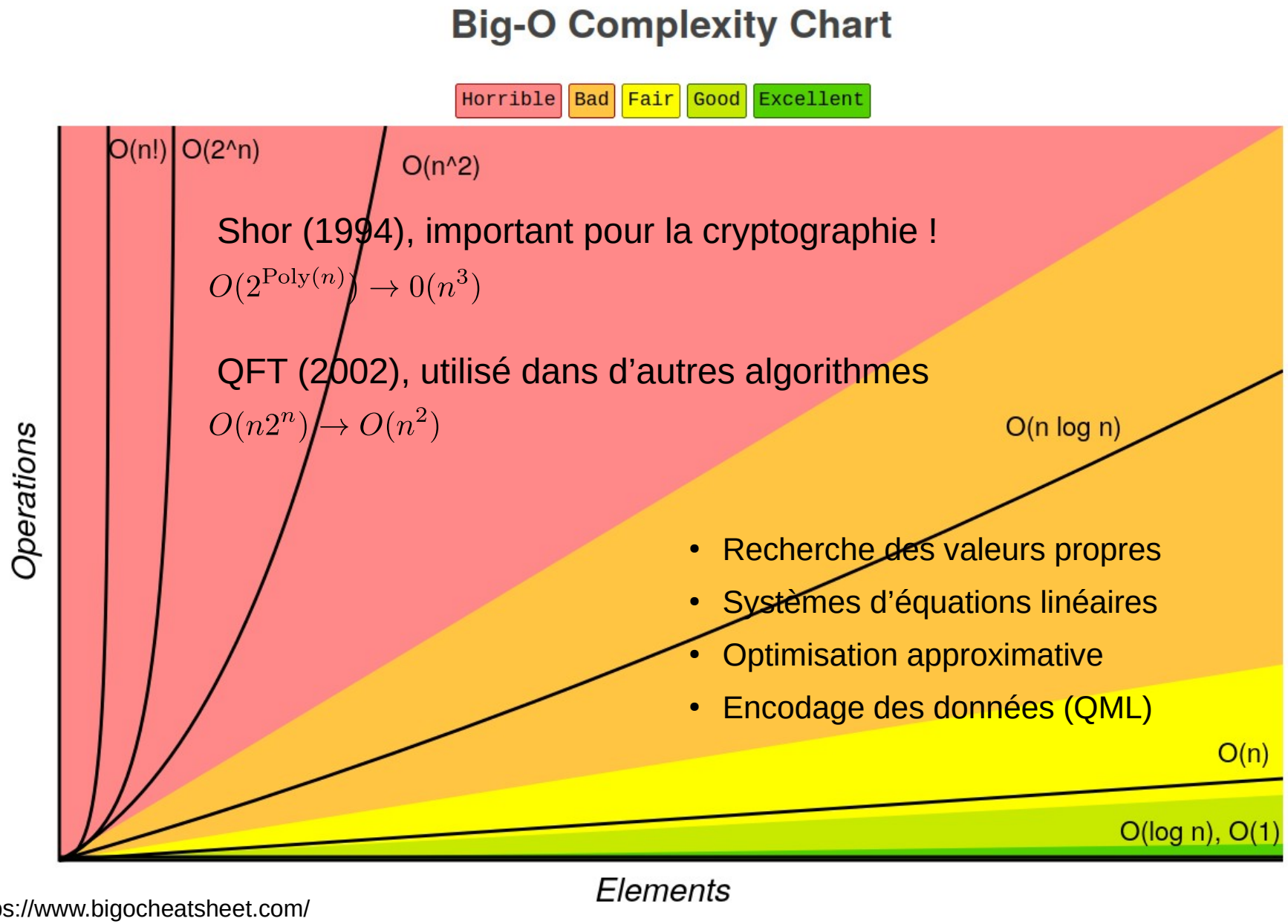
$O(2^{\mathrm{Poly}(n)}) \rightarrow 0(n^3)$

QFT (2002), utilisé dans d'autres algorithmes

$O(n2^n) \rightarrow O(n^2)$

Source https://www.bigocheatsheet.com/

Autres algorithmes



Shor (1994), important pour la cryptographie !

$O(2^{\mathrm{Poly}(n)}) \rightarrow 0(n^3)$

QFT (2002), utilisé dans d'autres algorithmes

$O(n2^n) \rightarrow O(n^2)$

- Recherche des valeurs propres
- Systèmes d'équations linéaires
- Optimisation approximative
- Encodage des données (QML)

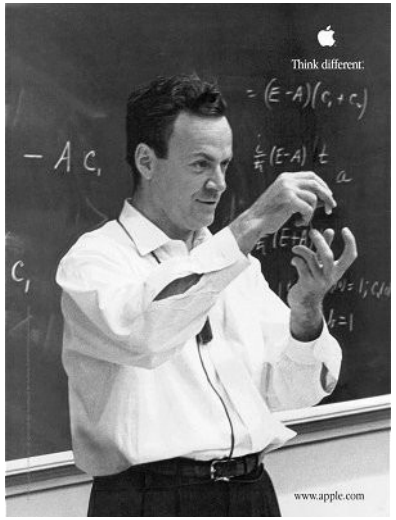
Source https://www.bigocheatsheet.com/

« Un **algorithme** est un ensemble de règles qui donne un résultat à partir d'une situation donnée. Chaque étape est soigneusement définie pour que sa traduction soit claire en langage informatique et réalisable par l'ordinateur. »

Donald Knuth, Pour la Science, juillet 1981, no 45

« Un **algorithme** est un ensemble de règles qui donne un résultat à partir d'une situation donnée. Chaque étape est soigneusement définie pour que sa traduction soit claire en langage informatique et réalisable par l'ordinateur. »
Donald Knuth, Pour la Science, juillet 1981, no 45

Il y a un type de processeur quantique qui peut résoudre des problèmes sans exécuter un vrai **algorithme**, mais plutôt en suivant la remarque faite par Richard Feynman (1918-1988) :



« Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical »

Un nombre de problèmes ont un **équivalent** dans des systèmes quantiques simples dont l'évolution peut être implémentée et mesurée avec un processeur quantique d'un certain type (la nature des qubits et de leur interconnections).

Des calculs de stratégie, ou opérationnels

Des calculs de stratégie, ou opérationnels

Solution

- Faire traverser la chèvre
- Retour
- Faire traverser le loup ou le chou
- Retourner avec la chèvre
- Faire traverser le chou ou le loup
- Retour
- Faire traverser la chèvre

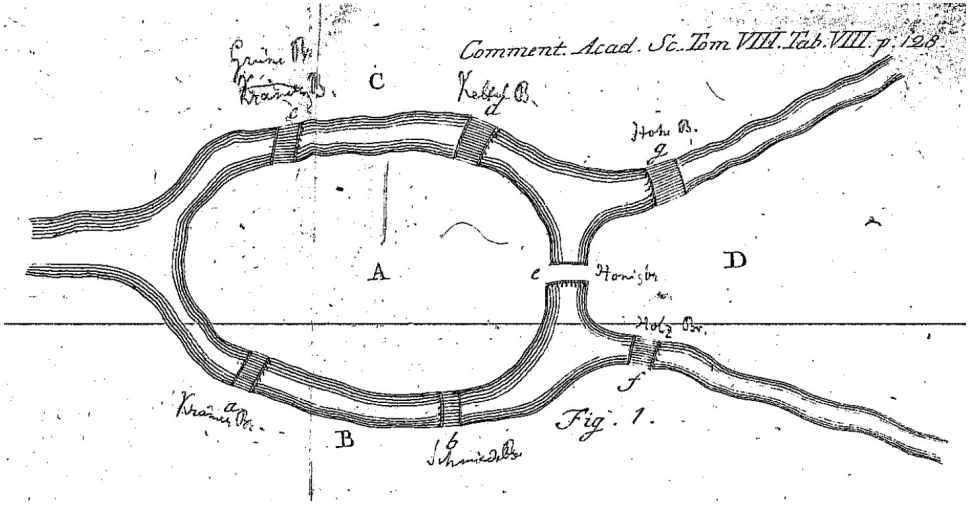
Sources : https://www.jardins-volpette.net/2017/09/menager-la-chevre-et-le-chou.html, Wikipedia

Les sept ponts de Koenigsberg (1736)

Leonhard Euler
1707-1783

Les sept ponts de Koenigsberg (1736)

Leonhard Euler
1707-1783



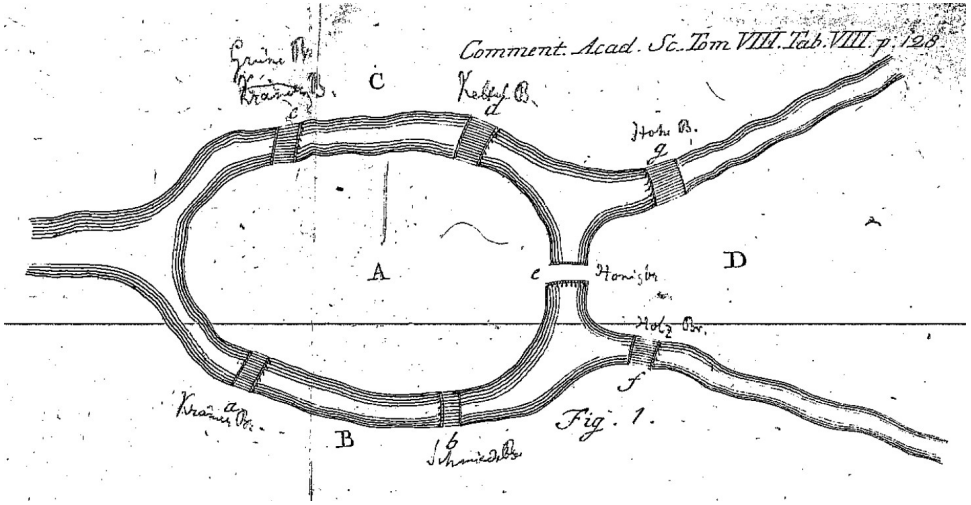
Solutio problematis ad geometriam situs pertinentis,

Leonhard Euler, 1741 http://eulerarchive.maa.org/

Existe-t-il un trajet, à partir d'un point au choix, qui **passe une seule fois par chacun des sept ponts et qui revient au point de départ** ?

Les sept ponts de Koenigsberg (1736)

Leonhard Euler
1707-1783



Solutio problematis ad geometriam situs pertinentis,
Leonhard Euler, 1741 http://eulerarchive.maa.org/

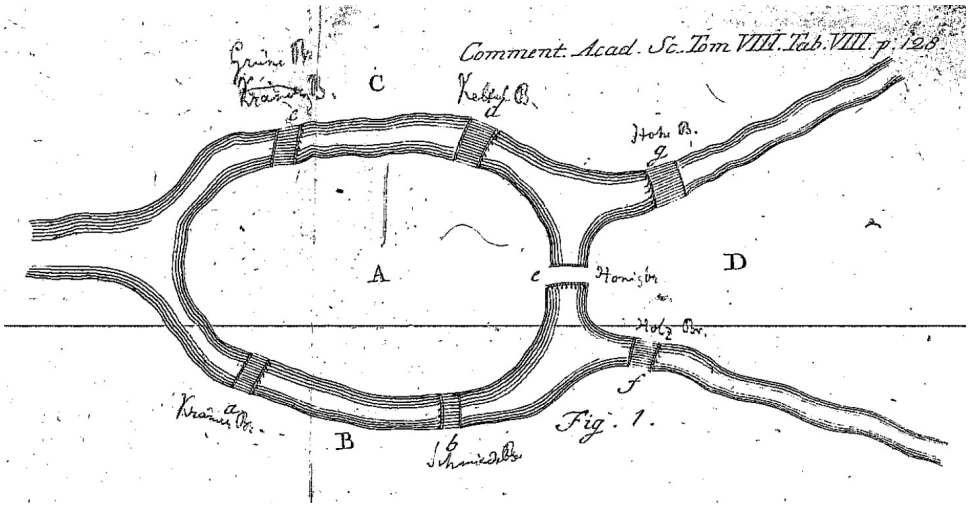
Existe-t-il un trajet, à partir d'un point au choix, qui **passe une seule fois par chacun des sept ponts et qui revient au point de départ** ?

Réponse : (dans ce cas) **NON**

Théorème de Euler : un graphe connexe admet un **cycle Eulérien** si et seulement si tous ses sommets sont de degré pair.

Les sept ponts de Koenigsberg (1736)

Leonhard Euler
1707-1783

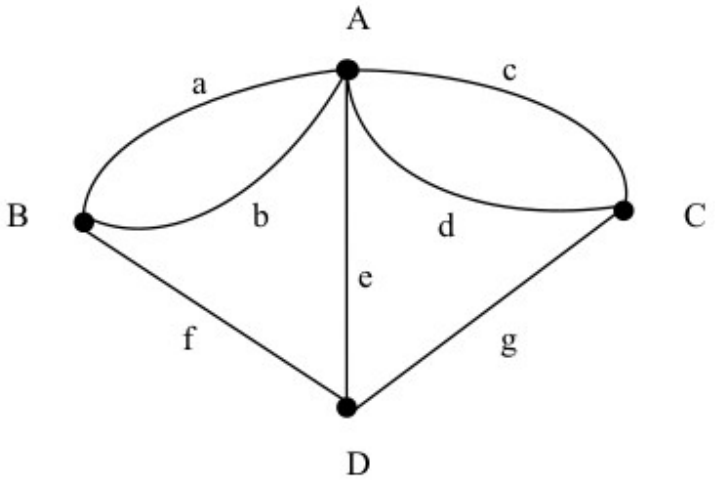


Solutio problematis ad geometriam situs pertinentis,
Leonhard Euler, 1741 http://eulerarchive.maa.org/

Existe-t-il un trajet, à partir d'un point au choix, qui **passe une seule fois par chacun des sept ponts et qui revient au point de départ** ?

Réponse : (dans ce cas) **NON**

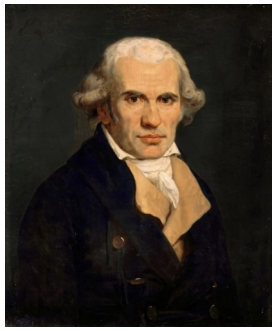
Théorème de Euler : un graphe connexe admet un **cycle Eulérien** si et seulement si tous ses sommets sont de degré pair.



Un **graphe** : une collection de points (nœuds, sommets) et de connexions entre eux (arêtes).

Un **parcours Eulérien** : un chemin qui passe par toutes Les arêtes une fois par arête.

Sources : Wikipedia



Gaspard Monge
1746-1818

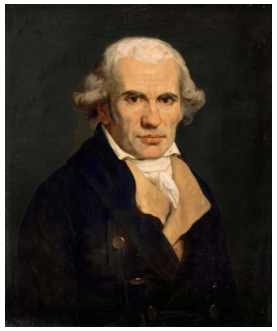
Gaspard Monge et la théorie du transport

1784

Rue à Clermont-Ferrand

Gaspard Monge et la théorie du transport

1784



Gaspard Monge
1746-1818

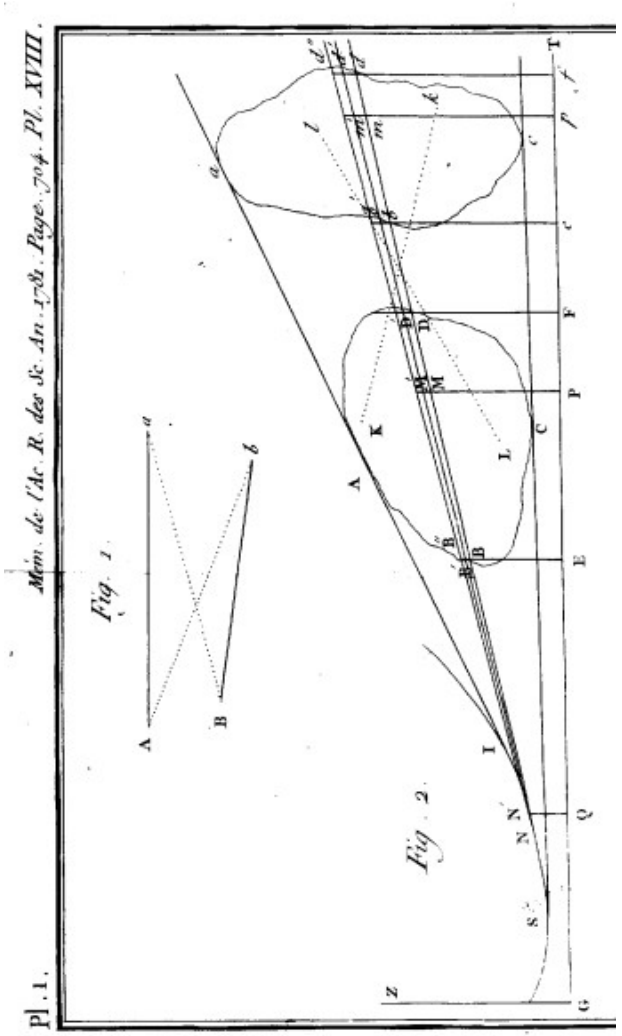
Rue à Clermont-Ferrand

MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.

Par M. MONGE.

LORSQU'ON doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit transporter, & le nom de *Remblai* à l'espace qu'elles doivent occuper après le transport.

Le prix du transport d'une molécule étant, toutes choses d'ailleurs égales, proportionnel à son poids & à l'espace qu'on lui fait parcourir, & par conséquent le prix du transport total devant être proportionnel à la somme des produits des molécules multipliées chacune par l'espace parcouru, il s'ensuit que le déblai & le remblai étant donnés de figure & de position, il n'est pas indifférent que telle molécule du déblai soit transportée dans tel ou tel autre endroit du remblai, mais qu'il y a une certaine distribution à faire des molécules du premier dans le second, d'après laquelle la somme de ces produits sera la moindre possible, & le prix du transport total sera un *minimum*.



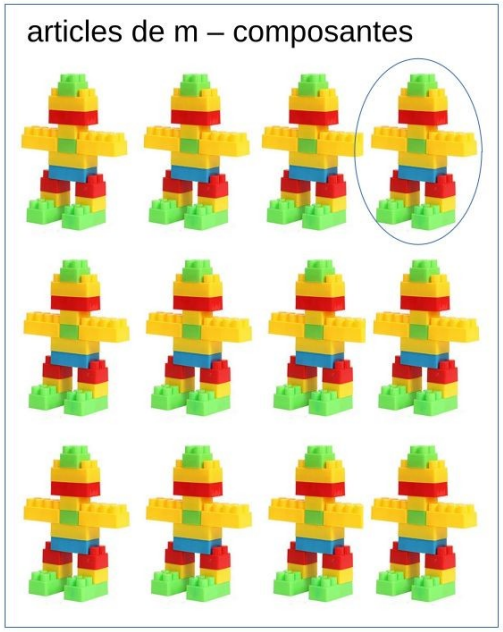
Sources : Wikipedia, maps.google

À l’aide du productivisme

À l’aide du productivisme

Leonid Kantorovich, 1939

Mathematical methods of organizing and planning production



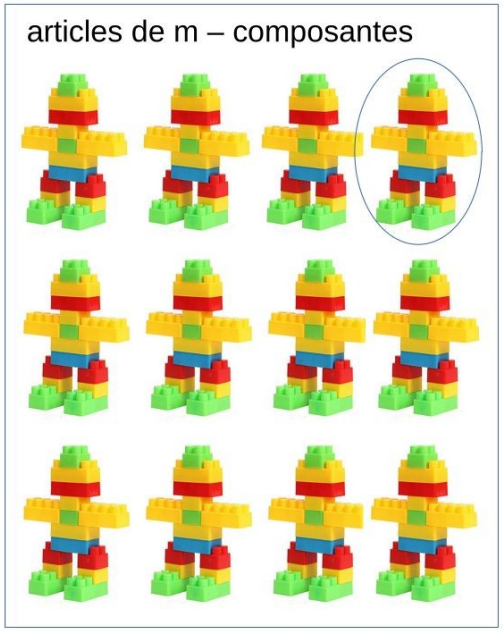
À l’aide du productivisme

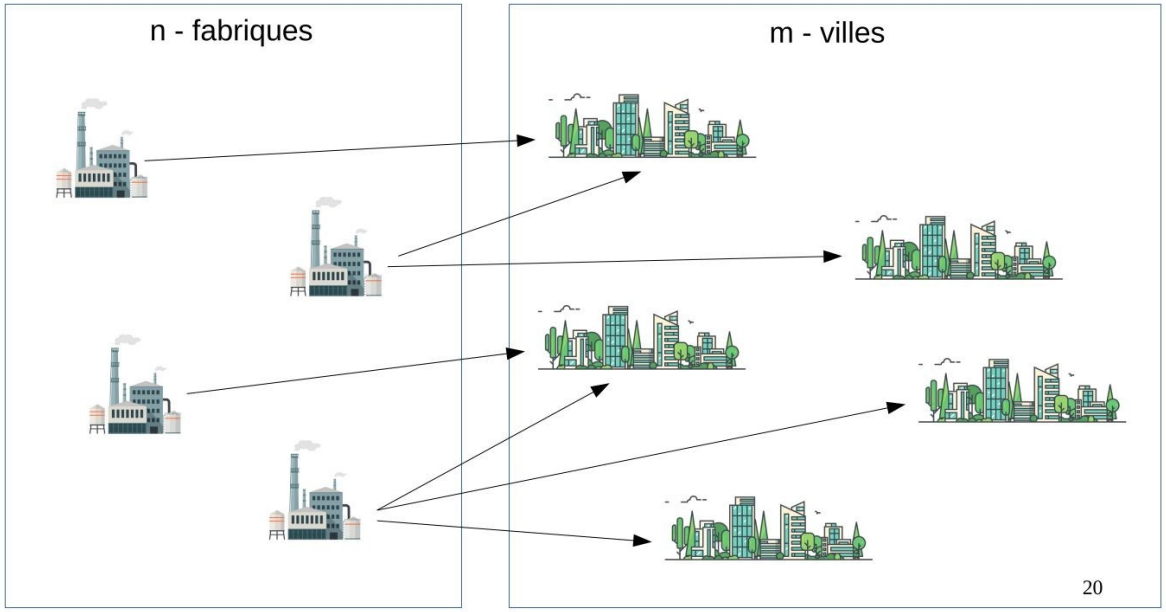
Leonid Kantorovich, 1939

Mathematical methods of organizing and planning production

Frank L. Hitchcock, 1941

The distribution of a product from several sources to numerous localities





La recherche opérationnelle et la programmation linéaire

George B. Dantzig, 1951

Maximization of a linear function of variables subject to linear inequalities

La recherche opérationnelle et la programmation linéaire

George B. Dantzig, 1951

Maximization of a linear function of variables subject to linear inequalities

PROBLEM: *Find the values of* $\lambda_1, \lambda_2, \cdots, \lambda_n$ *which maximize the linear form*

$$\lambda_1 c_1 + \lambda_2 c_2 + \cdots + \lambda_n c_n \tag{1}$$

subject to the conditions that

$$\lambda_j \geqq 0 \qquad (j = 1, 2, \cdots, n) \tag{2}$$

and

$$\begin{aligned} \lambda_1 a_{11} + \lambda_2 a_{12} + \cdots + \lambda_n a_{1n} &= b_1, \\ \lambda_1 a_{21} + \lambda_2 a_{22} + \cdots + \lambda_n a_{2n} &= b_2, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ &\cdot\ \cdot\ \cdot \\ \lambda_1 a_{m1} + \lambda_2 a_{m2} + \cdots + \lambda_n a_{mn} &= b_m, \end{aligned} \tag{3}$$

where a_{ij}, b_i, c_j *are constants* $(i = 1, 2, \cdots, m; j = 1, 2, \cdots, n)$.

Le problème du voyageur de commerce

(Travelling Salesman Problem, TSP)

Le problème du voyageur de commerce

(Travelling Salesman Problem, TSP)

[…] est un problème d'optimisation qui consiste à déterminer, étant donné un ensemble de villes et les distances entre toutes les paires de villes, un **plus court circuit qui passe par chaque ville une et une seule fois**.

Le problème du voyageur de commerce

(Travelling Salesman Problem, TSP)

[…] est un problème d'optimisation qui consiste à déterminer, étant donné un ensemble de villes et les distances entre toutes les paires de villes, un **plus court circuit qui passe par chaque ville une et une seule fois**.

⇒ la solution pour TSP : un **cycle Hamiltonien de longueur minimale** (coût minimal)

Le problème du voyageur de commerce

(Travelling Salesman Problem, TSP)

[…] est un problème d'optimisation qui consiste à déterminer, étant donné un ensemble de villes et les distances entre toutes les paires de villes, un **plus court circuit qui passe par chaque ville une et une seule fois**.

⇒ la solution pour TSP : un **cycle Hamiltonien de longueur minimale** (coût minimal)

Le nombre de cycles Hamiltoniens égal au **factoriel du nombre de villes**, si chaque ville est connectée par une route directe avec chaque autre ville, i.e. un graphe **complet** = tout les sommets sont adjacents deux à deux. $(\text{le factoriel de } N \text{ est le produit } 1 \cdot 2 \cdot 3 \cdot \ldots \cdot N)$

Le problème du voyageur de commerce

(Travelling Salesman Problem, TSP)

[…] est un problème d'optimisation qui consiste à déterminer, étant donné un ensemble de villes et les distances entre toutes les paires de villes, un **plus court circuit qui passe par chaque ville une et une seule fois**.

⇒ la solution pour TSP : un **cycle Hamiltonien de longueur minimale** (coût minimal)

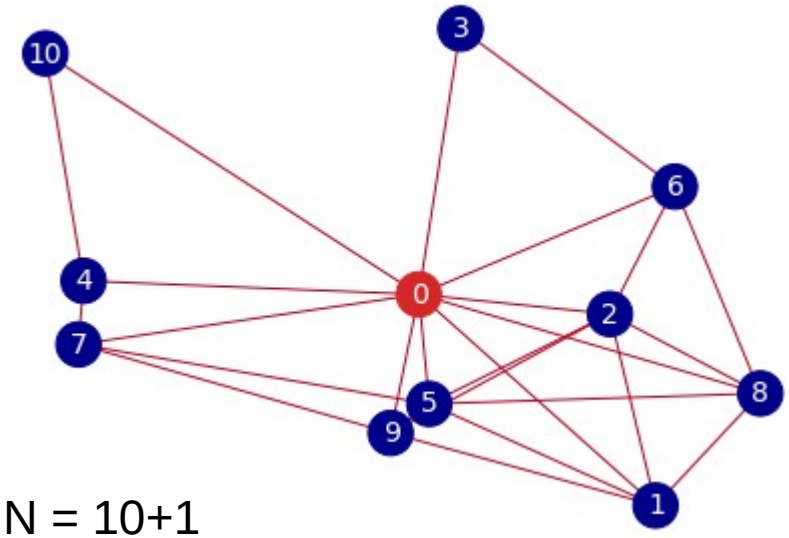
Le nombre de cycles Hamiltoniens égal au **factoriel du nombre de villes**, si chaque ville est connectée par une route directe avec chaque autre ville, i.e. un graphe **complet** = tous les sommets sont adjacents deux à deux. $(\text{le factoriel de } N \text{ est le produit } 1 \cdot 2 \cdot 3 \cdot \ldots \cdot N)$

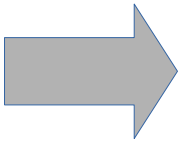
On ne connaît pas d'algorithme permettant de trouver une solution exacte rapidement dans tous les cas. Plus précisément, on ne connaît pas d'algorithme en **temps polynomial**.

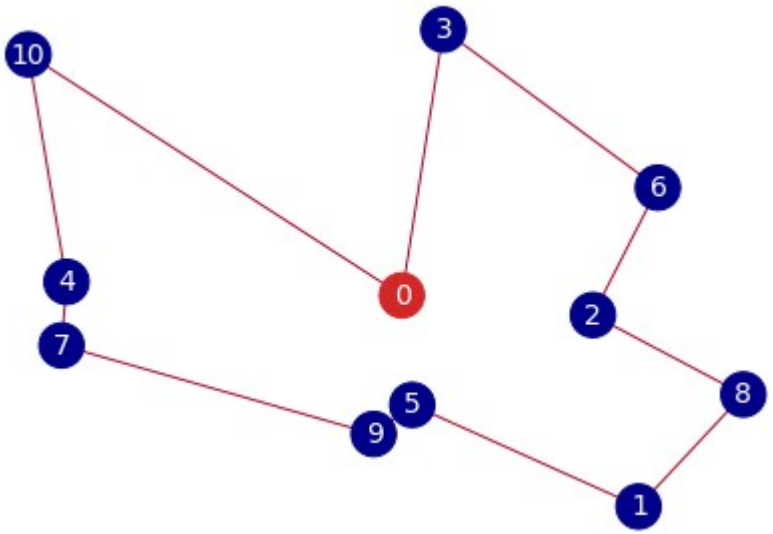
Source : Wikipedia, théorie de la complexité

Exemples de TSP avec leurs solutions optimales

Exemples de TSP avec leurs solutions optimales

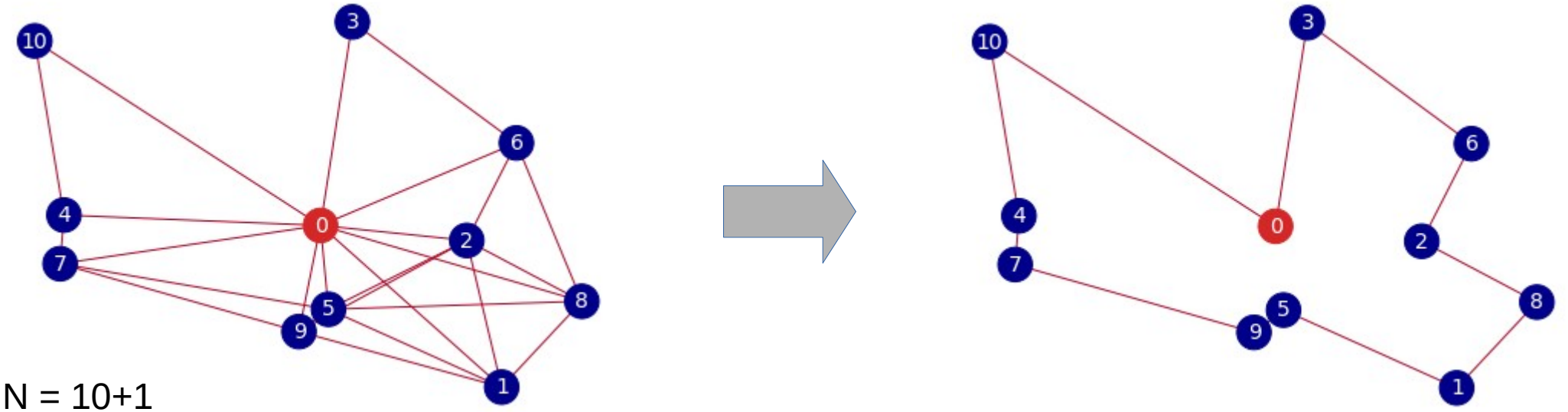




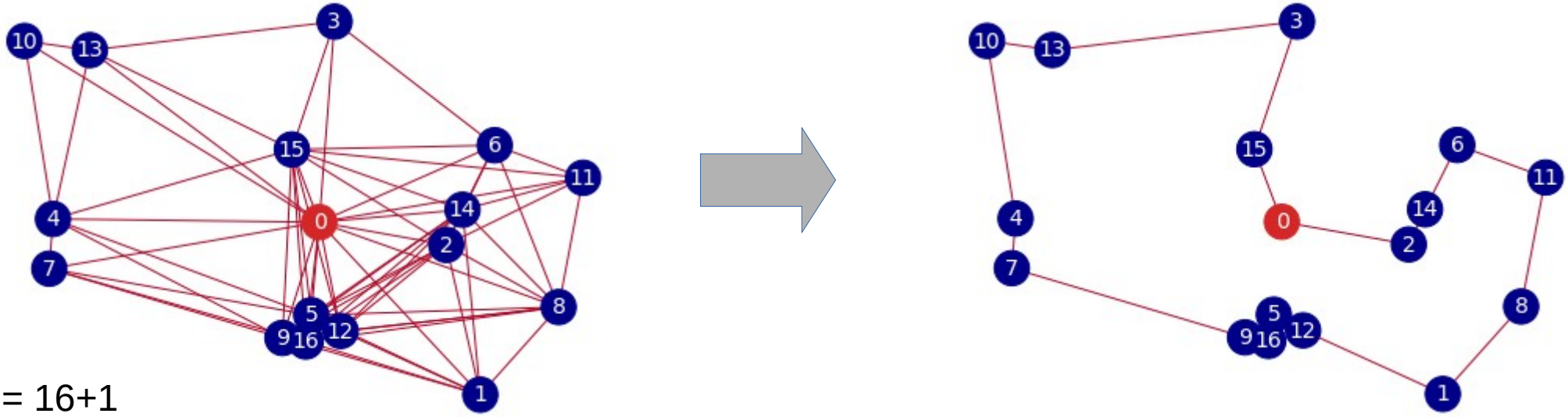


N = 10+1

Exemples de TSP avec leurs solutions optimales



N = 10+1



N = 16+1

Solutions avec IBM CPLEX

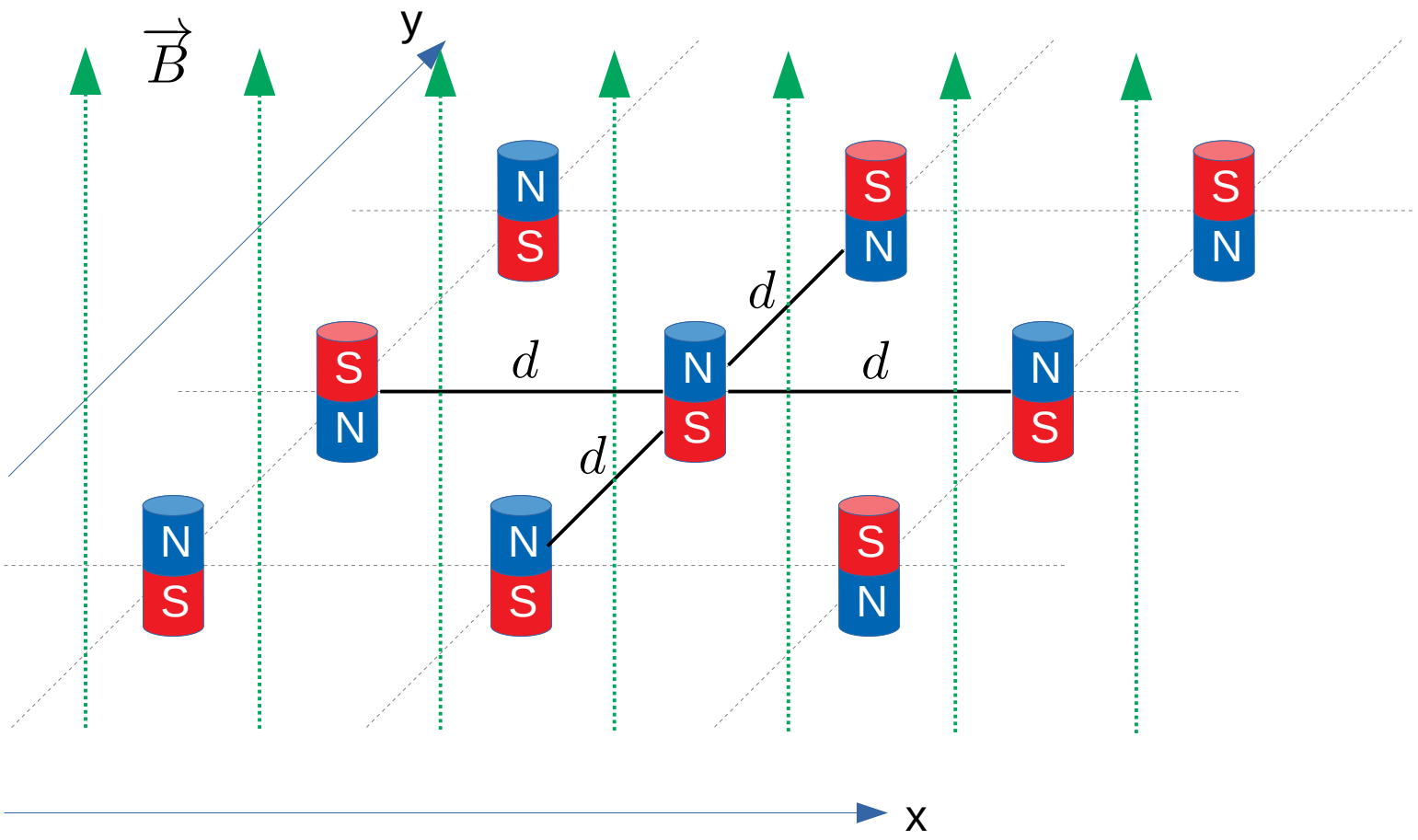
Un autre type de processeur quantique

Un autre type de processeur quantique

un ensemble de qubits dans un réseaux décrit par le **modèle Ising**
plus
une conséquence du **théorème quantique adiabatique**

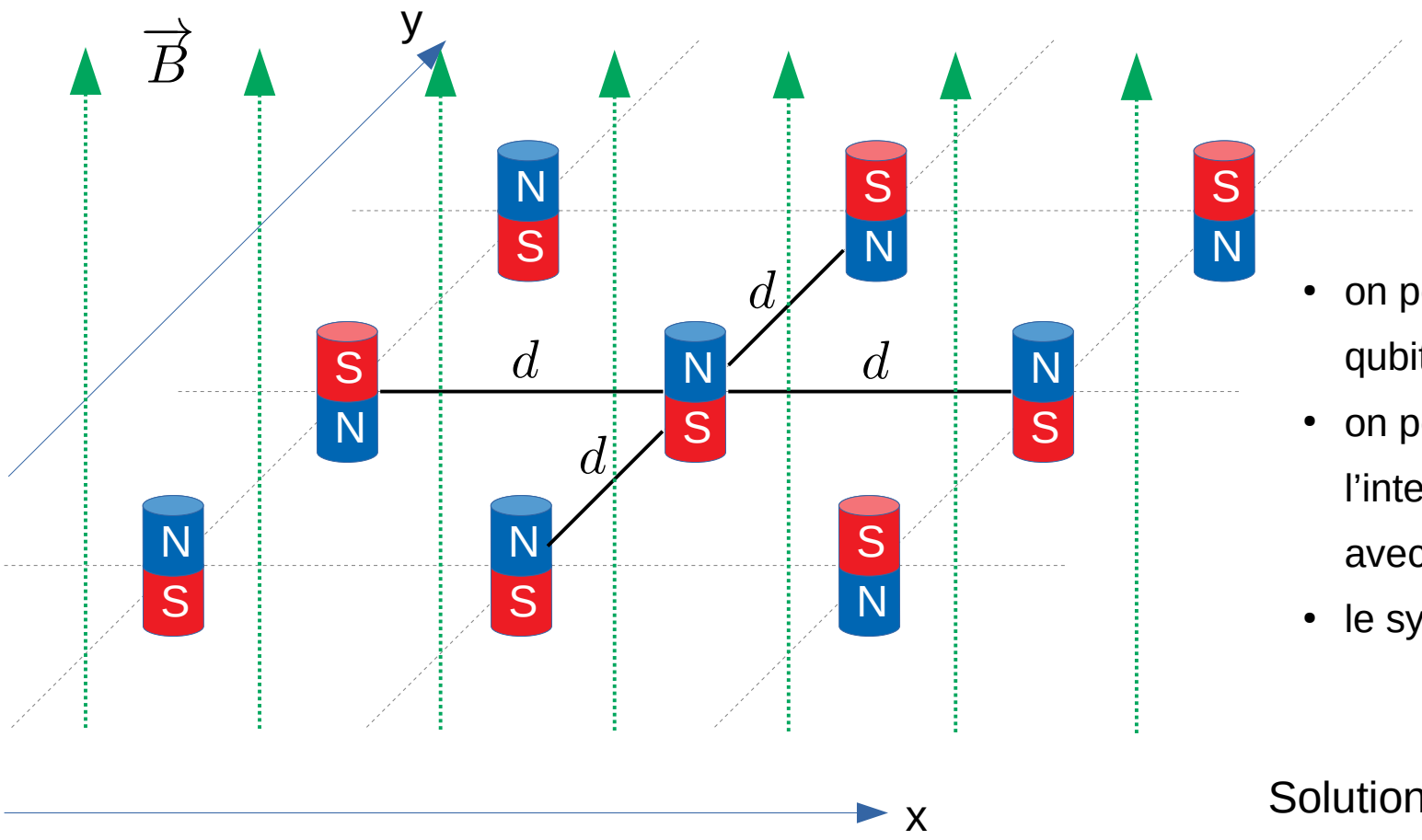
Un autre type de processeur quantique

un ensemble de qubits dans un réseaux décrit par le **modèle Ising**
plus
une conséquence du **théorème quantique adiabatique**



Un autre type de processeur quantique

un ensemble de qubits dans un réseaux décrit par le **modèle Ising**
plus
une conséquence du **théorème quantique adiabatique**



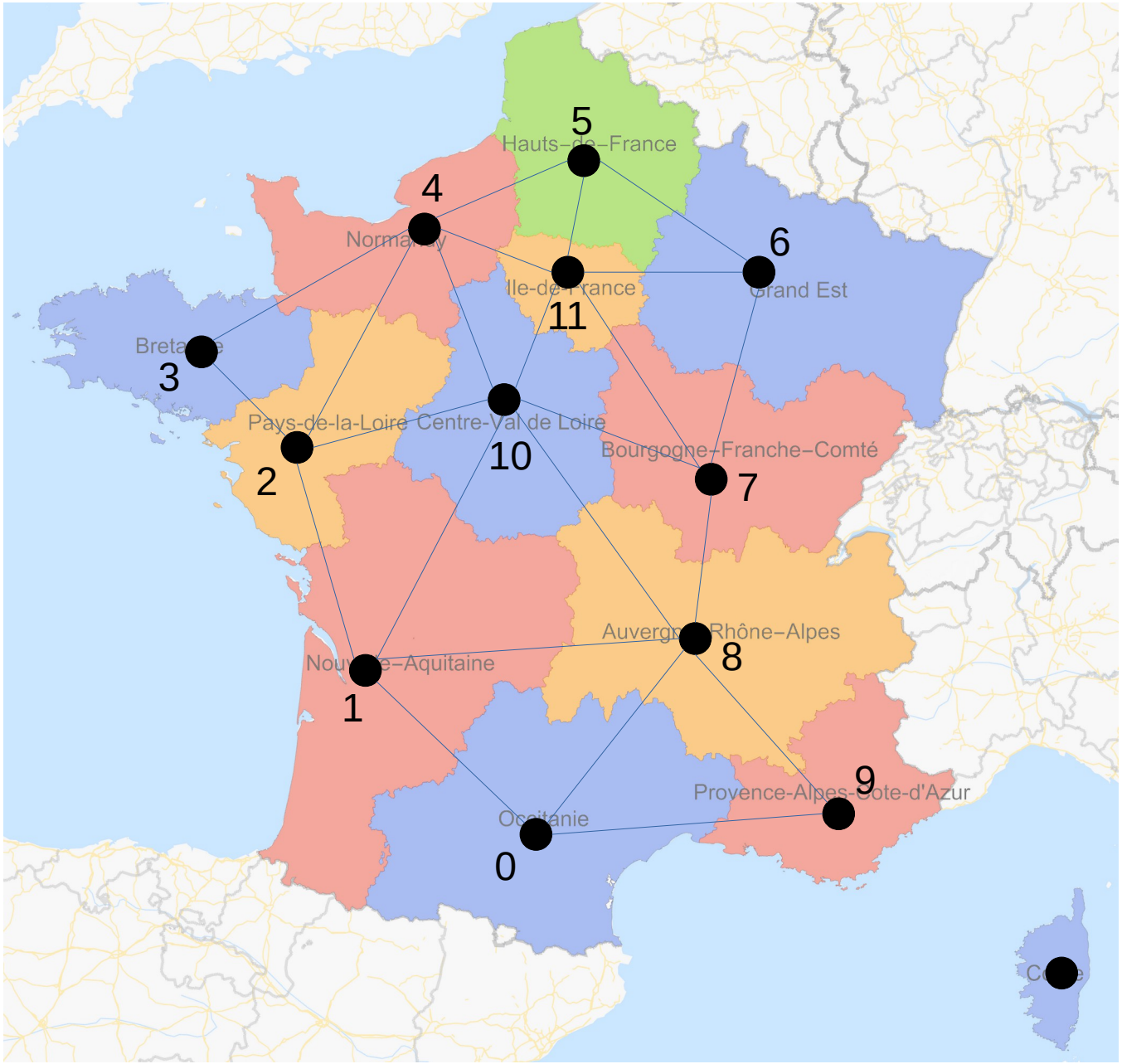
Paramètres du problème

- on peut contrôler l'état initial des qubits
- on peut contrôler l'intensité de l'interaction entre les qubits et avec un champ de force externe
- le système est libre d'évoluer

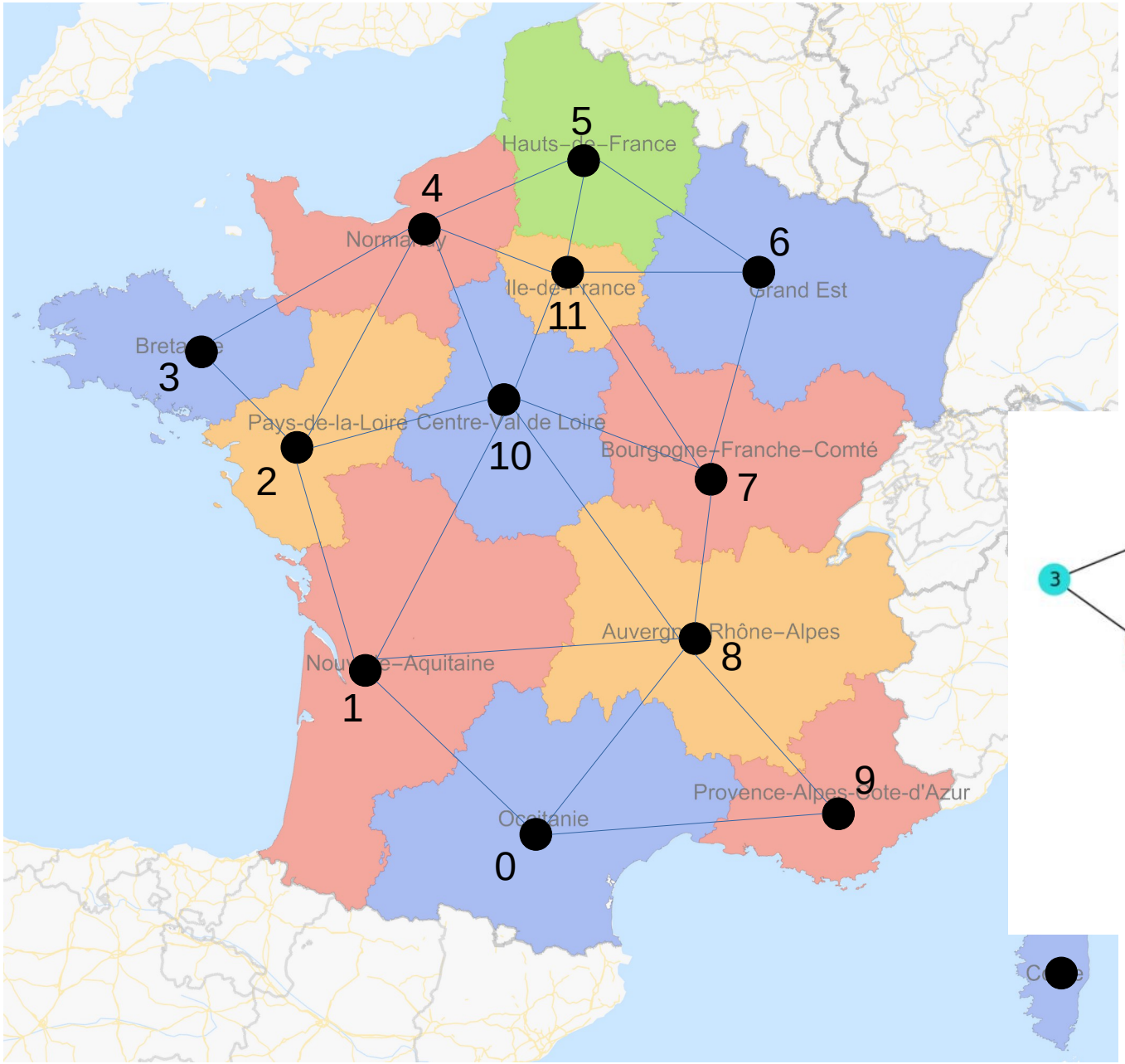
Solution(s) du problème

Le problème du coloriage (avec quatre couleurs)

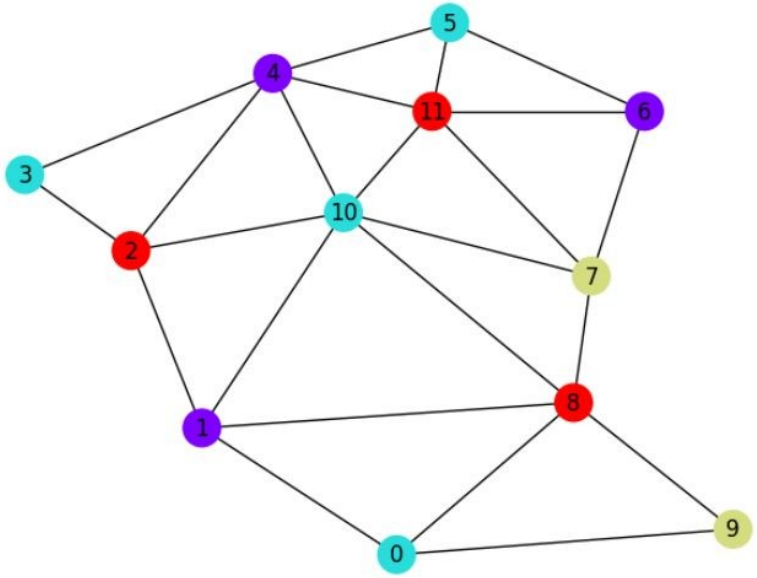
Le problème du coloriage (avec quatre couleurs)



Le problème du coloriage (avec quatre couleurs)



Solution avec le **modèle quadratique discret** de la bibliothèque de programmation pour le processeur quantique de l'entreprise **DWave** (CA).



Les plus puissants superordinateurs (classiques) au monde

Les plus puissants superordinateurs (classiques) au monde

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,340,000	1,809.00	2,821.10	29,685
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
3	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
4	**JUPITER Booster** - BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux, EVIDEN EuroHPC/FZJ Germany	4,801,344	1,000.00	1,226.28	15,794
5	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	

Sources : https://top500.org/lists/top500/2025/11/

Les plus puissants superordinateurs (classiques) au monde

TSP par **force brute** :

$$2\ \mathrm{PFlop/s} = 2 \cdot 10^{15}\ \mathrm{Flop/s}$$

$$1\mathrm{y} = 3.1536 \cdot 10^{7}\mathrm{s}$$

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,340,000	1,809.00	2,821.10	29,685
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
3	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
4	**JUPITER Booster** - BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux, EVIDEN EuroHPC/FZJ Germany	4,801,344	1,000.00	1,226.28	15,794
5	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	

Sources : https://top500.org/lists/top500/2025/11/

Les plus puissants superordinateurs (classiques) au monde

TSP par **force brute** :

$$2\ \mathrm{PFlop/s} = 2 \cdot 10^{15}\ \mathrm{Flop/s}$$

$$1\mathrm{y} = 3.1536 \cdot 10^{7}\mathrm{s}$$

$$30! \simeq 2.65 \cdot 10^{32}$$

$$4.2 \cdot 10^{9}\mathrm{y}$$

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,340,000	1,809.00	2,821.10	29,685
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
3	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
4	**JUPITER Booster** - BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux, EVIDEN EuroHPC/FZJ Germany	4,801,344	1,000.00	1,226.28	15,794
5	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	

Sources : https://top500.org/lists/top500/2025/11/

Les plus puissants superordinateurs (classiques) au monde

TSP par **force brute** :

$$2\ \mathrm{PFlop/s} = 2 \cdot 10^{15}\ \mathrm{Flop/s}$$

$$1\mathrm{y} = 3.1536 \cdot 10^{7}\mathrm{s}$$

$$30! \simeq 2.65 \cdot 10^{32}$$

$$4.2 \cdot 10^{9}\mathrm{y}$$

$$100! \simeq 9.33 \cdot 10^{157}$$

$$1.5 \cdot 10^{135}\mathrm{y}$$

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,340,000	1,809.00	2,821.10	29,685
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
3	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
4	**JUPITER Booster** - BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux, EVIDEN EuroHPC/FZJ Germany	4,801,344	1,000.00	1,226.28	15,794
5	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	

Sources : https://top500.org/lists/top500/2025/11/

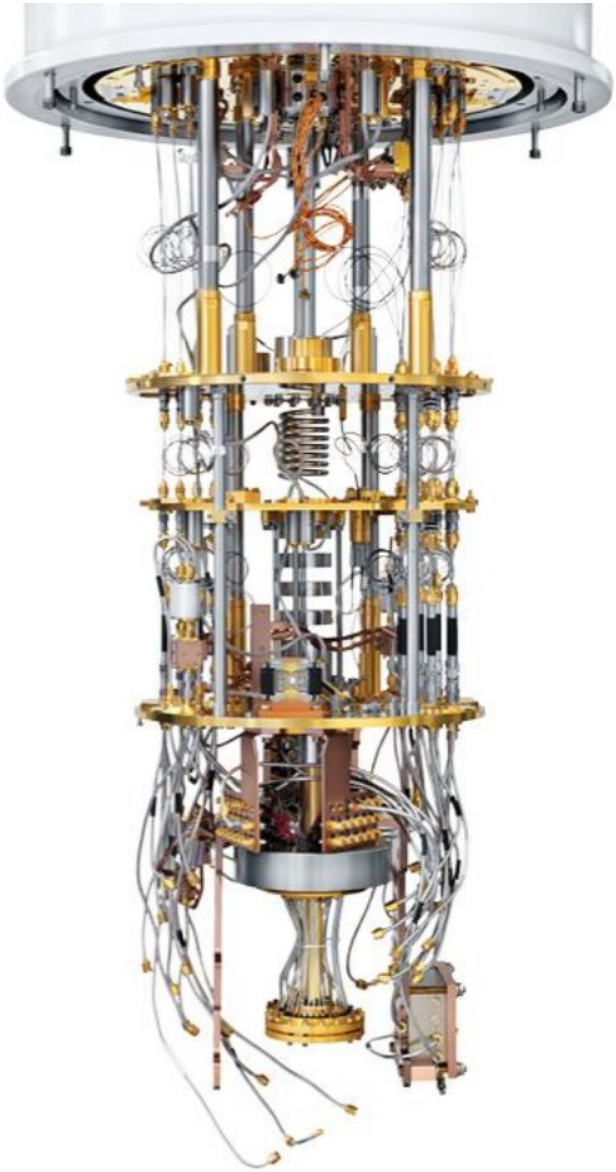
Processeurs quantiques à portes (implémentation d’algorithmes)

Processeurs quantiques à portes (implémentation d’algorithmes)

IBM

Processeurs quantiques à portes (implémentation d'algorithmes)

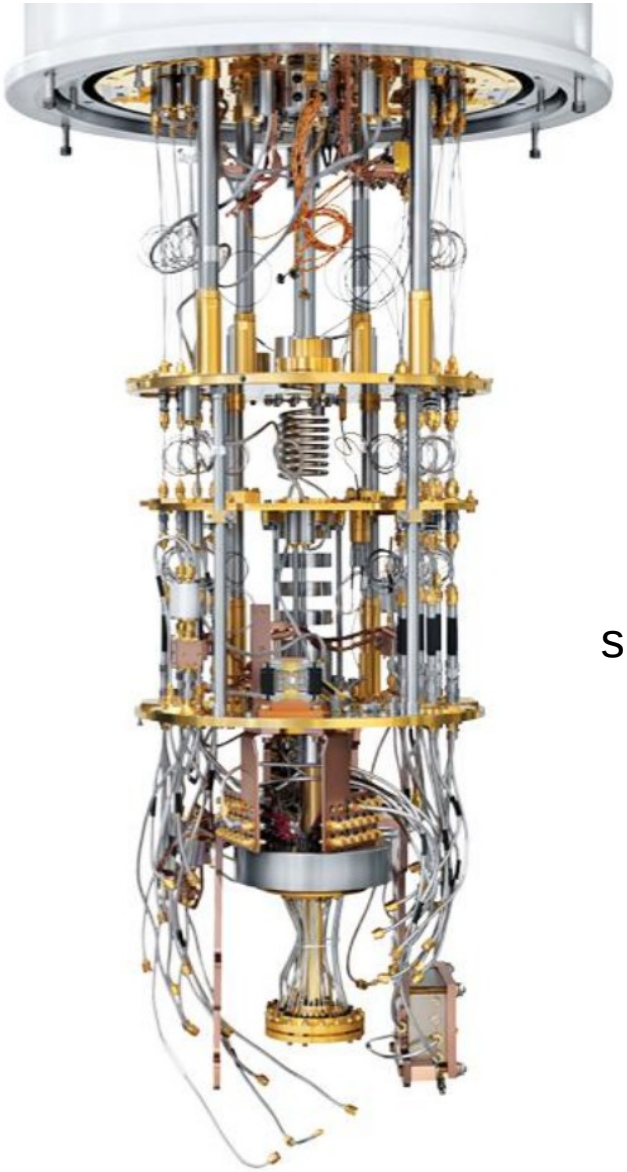
IBM



Rigetti

Processeurs quantiques à portes (implémentation d'algorithmes)

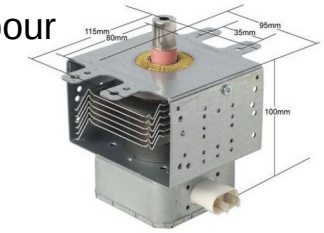
IBM



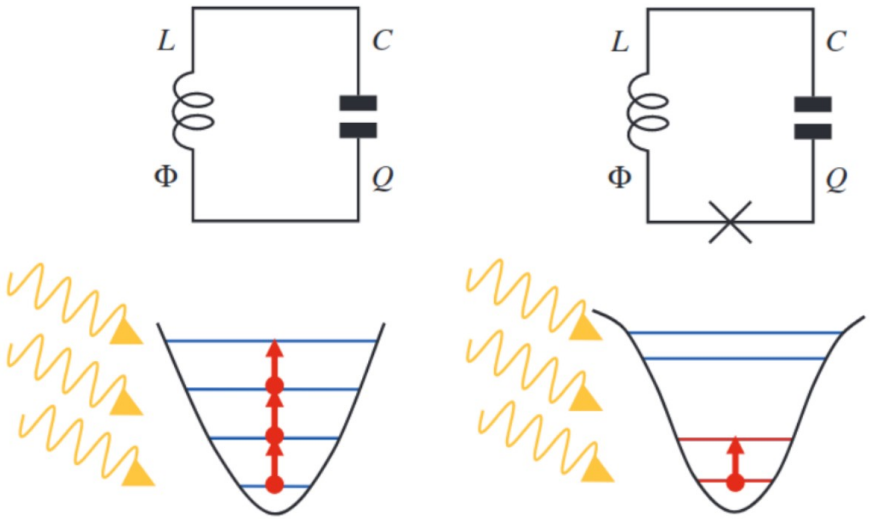
Rigetti

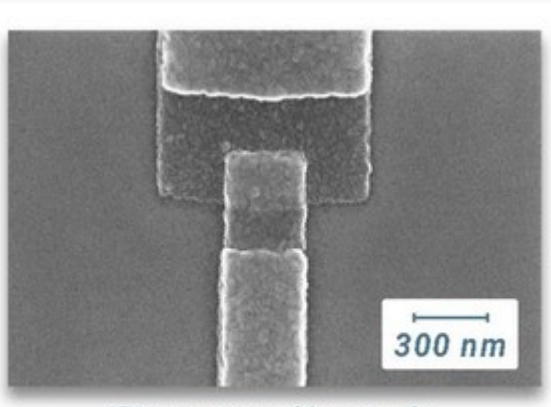
> 200 qubits supra-conducteurs fonctionnant à des températures ~ 1 mK (proche de zéro absolu -273.15 Celsius) et manipulés avec des signaux dans le domaine des micro-ondes (GHz)

Magnetron pour le four à micro-ondes

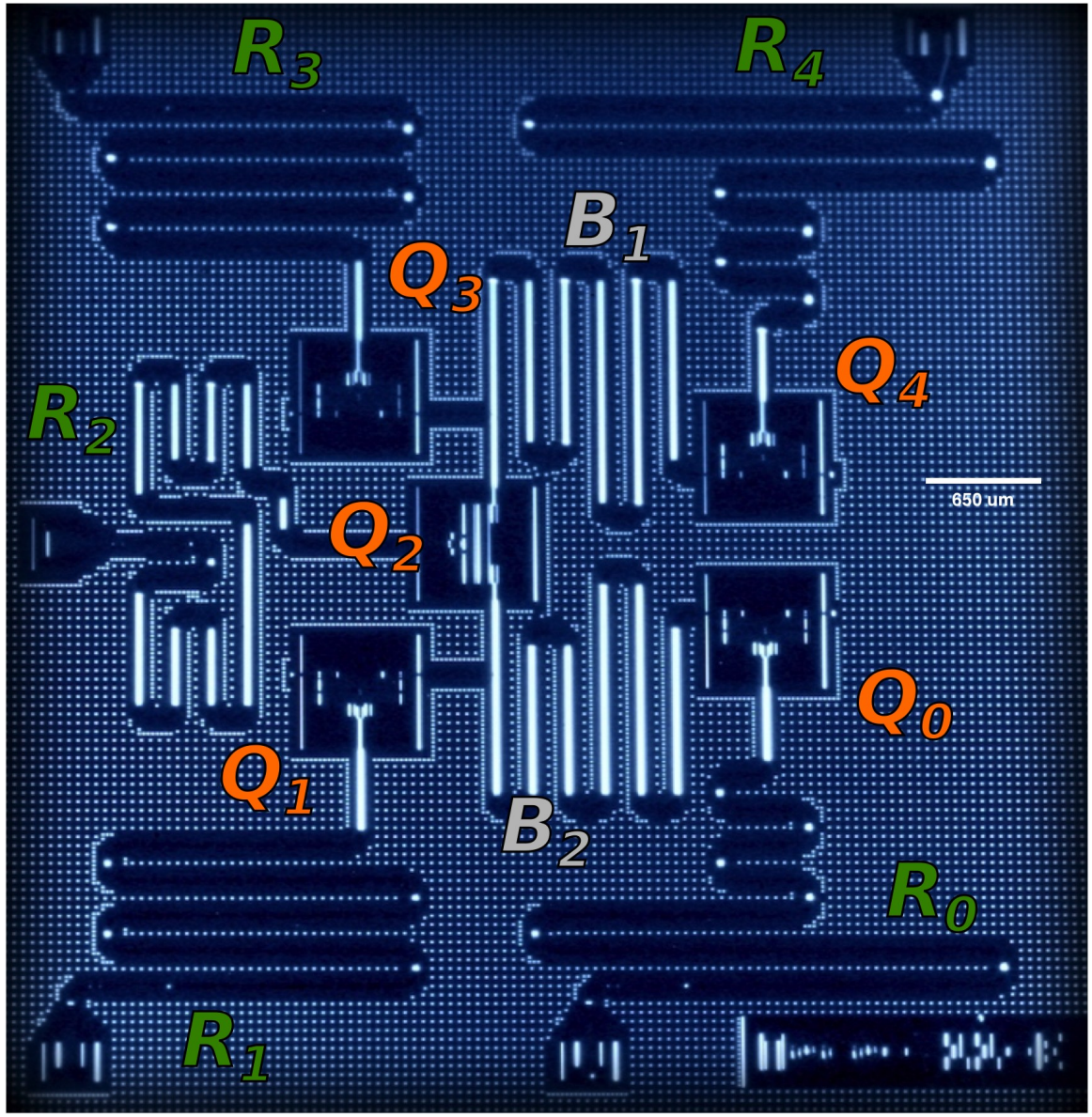


Boucle de courent supraconducteur oscillant, la jonction Josephson et l'effet tunnel : un système quantique macroscopique.





SEM image courtesy of the Institute for Quantum Computing (IQC) at the University of Waterloo



Un précurseur : le processeur IBM QX2 (5 qubits)

QUANDELA (FR)

processeur quantique **photonique** (24 photons)

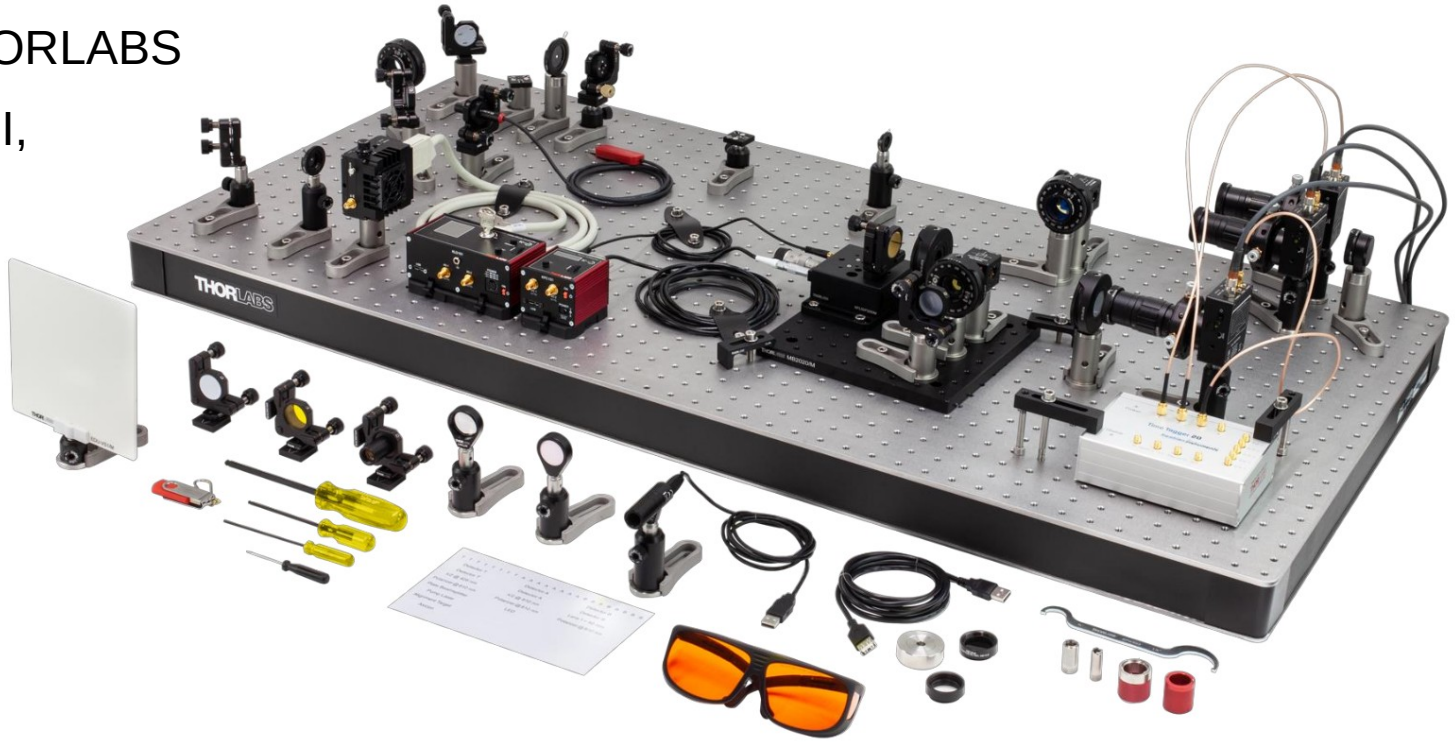
Lucy @ GENCI/TGCC (CEA)

QUANDELA (FR)

processeur quantique **photonique** (24 photons)

Lucy @ GENCI/TGCC (CEA)

Kit d’optique quantique THORLABS pour les étudiants de l’EUPI, Master Nanophysique et cours de **Quantum Computing** (rentrée 2026, avec l’Institut Pascal)



CEA TGCC Bruyère-le-Châtel

Processeurs quantiques analogiques

Processeurs quantiques analogiques

DWave (CA)

> 5000 qubits supra-conducteurs

Processeurs quantiques analogiques

DWave (CA)

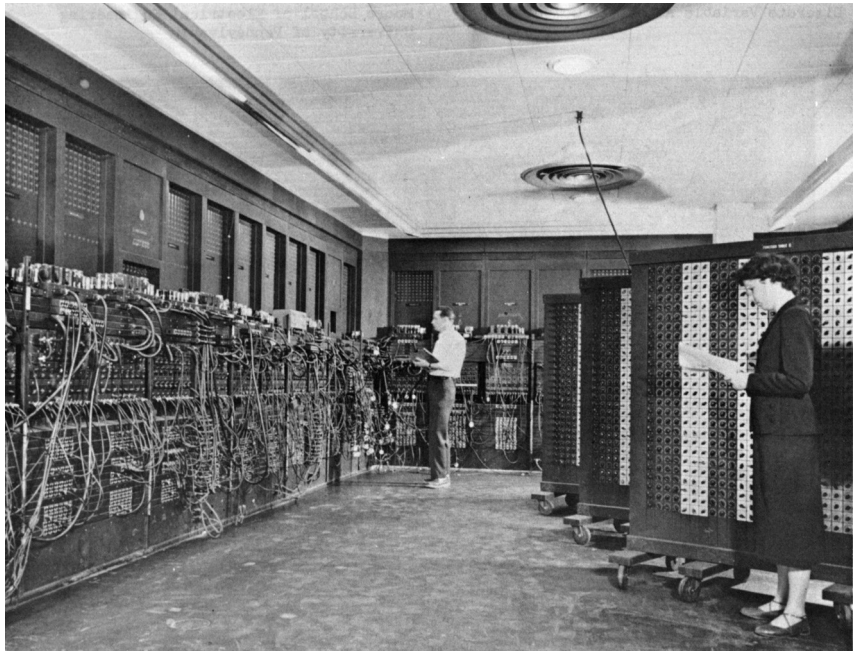
> 5000 qubits supra-conducteurs

PASQAL (FR)

100 qubits atomes neutres (Rubidium)

@ GENCI/TGCC (CEA)

Merci de votre attention !



Ordinateur classique, Eniac ~1950

Ordinateur quantique, IBM ~2020