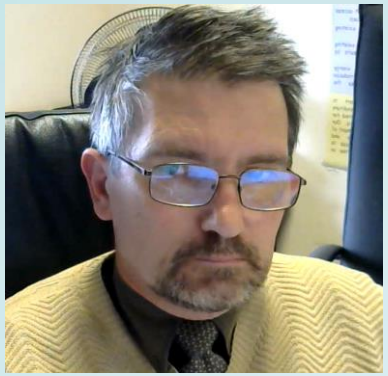
Introduction à l'Informatique Quantique

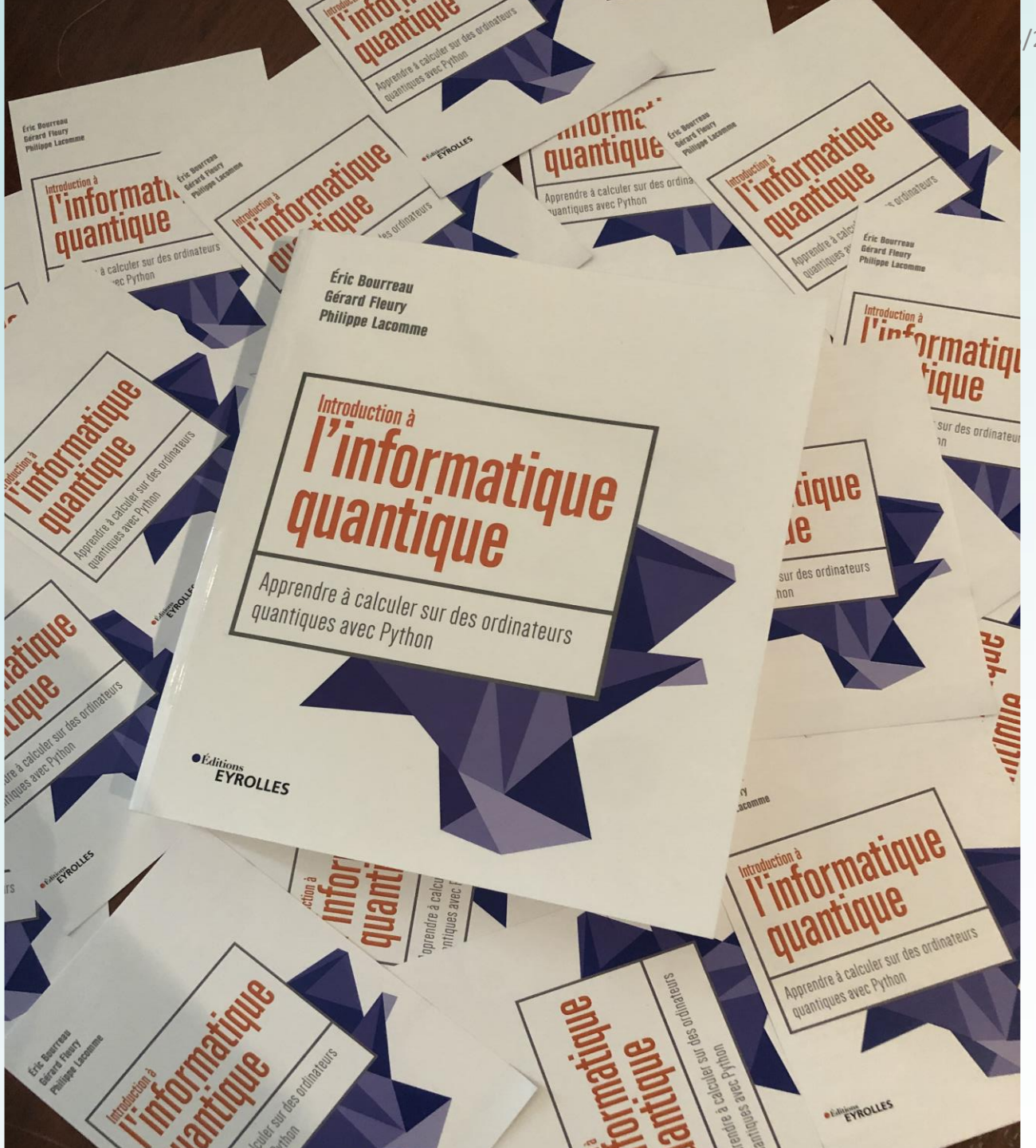
Eric Bourreau, LIRMM (Laboratoire d'Informatique Robotique et Microélectronique de Montpellier)

Université de Montpellier



Philippe Lacomme
(LIMOS)

Gérard Fleury
(LIMOS)

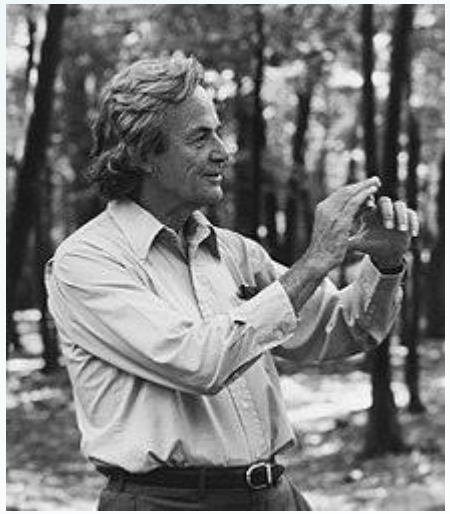


Contexte

- GDR RO → Recherche Opérationnelle Quantique
- Introduction
  - Concepts
  - Machines / Technologies
  - Algorithmes
- Algo quantique avec Grover sur un problème NP-complet
  - Problème
  - Modèle
  - Code et résolution live
- Conclusions/Perspectives

Informatique Quantique

- Un ordinateur quantique
  - Richard Feynmann
    Simulating Physics with Computers, Int. J. Theor. Physics, vol 21, n°6/7, 1982, pp 471-493
    « *…a place where the relationship of physics and computation has turned itself the other way and told us something about the possibilities of computation …* »
    « *Can you do it with a new kind of computer--a quantum computer?* »



- Des phénomènes étranges
  - **Superposition**
  - **Intrication**
  - **Fragilité d'observation**





De la théorie à la pratique

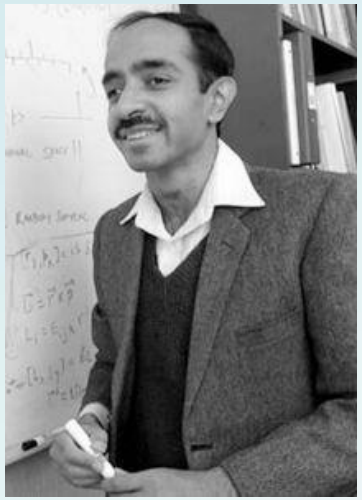
Zuchongzhi

1982 Feynmann → 1994 Shor → 1996 Grover

→ 2000 : 5 Qubits computer → 2011 Dwave computer

→ 2019 Google Quantum Supremacy

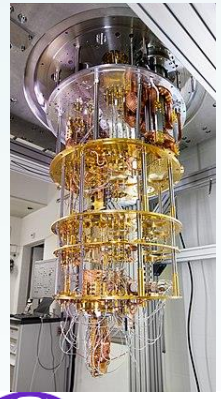




IBM

rigetti

D:wave

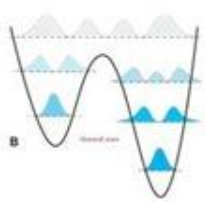
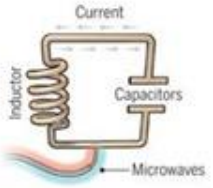
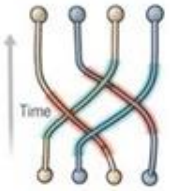
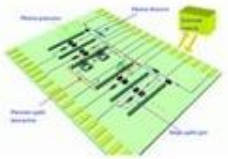
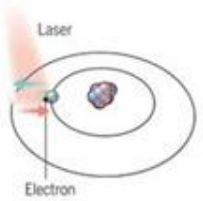
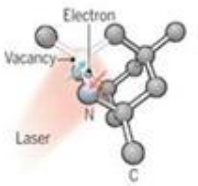


IONQ

Qiskit

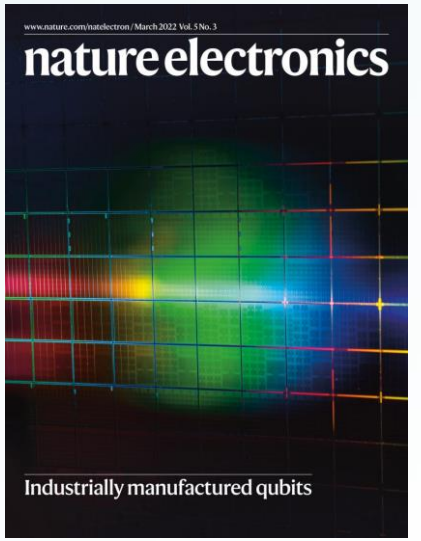
https://en.wikipedia.org/wiki/Timeline_of_quantum_computing_and_communication

Les machines

							
	recuit quantique	**boucles supra-conductrices**	**qubits topologiques**	**optique linéaire**	**quantum dots silicium**	**ions piégés**	**cavités diamants**
qubit	supraconducteur effet Josephson	supraconducteur effet Josephson	quasi-particules faites de paires d'anyons	photons	spin d'électrons dans semi-conducteur	ions piégés magnétiquement	spin de noyau d'atomes
# qubit	**2048 qubits** (D-Wave)	**50 qubits** (IBM) **72 qubits** (Google)	**N/A**	**quelques-uns**	**49 qubits** (Intel)	**53 qubits** (IonQ) **51 qubits** (MIT) **20 qubits** (IQOQI)	**6 qubits** (QDTI)
état	sens du courant	phase de résonnance ou sens du courant	sens de l'anyon	phase de photon	spins d'électrons	niveau énergétique de l'ion piégé	niveau d'énergie de la cavité
portes	micro-ondes 5 GHz et effet Josephson	micro-ondes 5 GHz et effet Josephson	inversions 2D d'anyons	filtres polarisants et dichroïques	micro-ondes	laser	laser
mesure	magnétomètre	magnétomètre	fusion d'anyons	détecteurs de photons	consersion spins to charge	fluorescence	fluorescence

Source : frenchweb.fr

- **Pasqal : atomes froids, 100+ qbits**



Les start-ups

Les algorithmes (pour l'optimisation combinatoire)

- Algorithme de Grover (1996) :
  - Trouver une solution marquée par un Oracle parmi N états
  - Recherche complète
  - $O(\sqrt{N})$ … donc pour un espace combinatoire en 2^N =>$\sqrt{2^N}$
  - Le TSP en DynProg à la Held&Carp ➔ améliore la meilleure complexité connue $O^*(1{,}728^n)$

    Quantum Speedups for Exponential-Time Dynamic Programming Algorithms, Ambainis et al, 2018

- Recuit Quantique et Optimisation Adiabatique (Farhi 2000) :
  - Utilisation de la descente spontanée (ou simulée) de l'énergie d'un système observable
  - Simplification d'un problème d'optimisation sous la forme d'un hamiltonien (QUBO)
  - Preuve de convergence théorique

- VQE (forme variationnelle) et QAOA (approximation) (Farhi 2014) :
  - Représentation d'un problème sous la forme d'un Hamiltonien
  - Simplification de la descente adiabatique par une approche variationnelle (on teste des angles et on recommence)
  - Méthode hybride (besoin d'un méta optimiseur black box sur ordinateur classique)

Jeudi 12 mai
11h00-11h45

Jeudi 12 mai
14h00-15h30

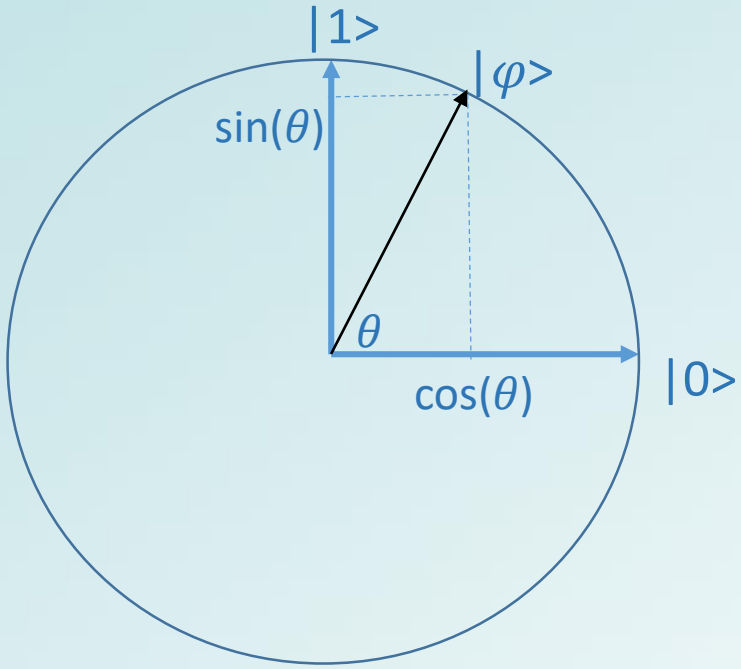
Jeudi 12 mai
16h-17h30

Vendredi 13 mai
10h30-12h30

Vendredi 13 mai
9h00-10h00

Vendredi 13 mai
14h00-16h30

Qubit



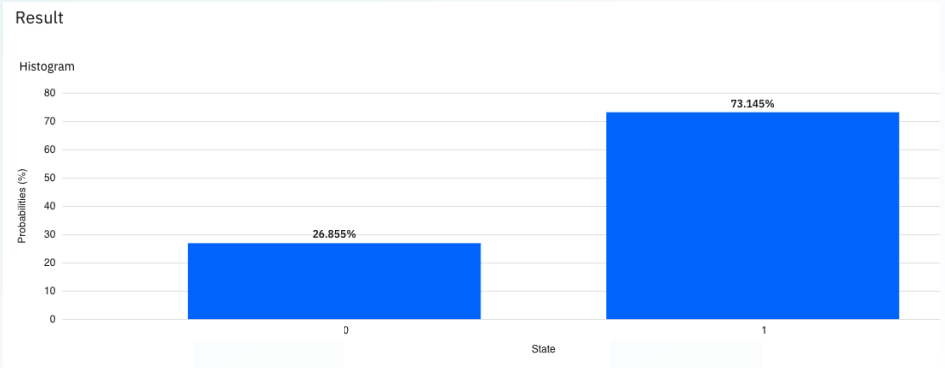
- Définition théorique :
  Un bit quantique (ou QuBit) est un vecteur de norme 1 dans l'espace canonique $\mathbb{C}^2$ de Hilbert
- Les bases sont :

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ et } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- Son vecteur d'état est une combinaison linéaire entre les deux états |0> et |1>

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \text{ avec } |\alpha|^2 + |\beta|^2 = 1, \alpha \text{ et } \beta \in \mathbb{C}$$

(La représentation graphique ne considère ici que α et $\beta \in \mathbb{R}$)

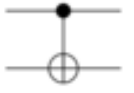
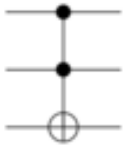


- Un qubit peut être observé uniquement dans un état aléatoirement choisi entre |0> ou |1> avec une probabilité proportionnelle à son amplitude au carré α^2 et β^2
- Une fois mesuré un qubit est projeté définitivement dans un état (il s'écroule – *collapse*) et détruit toute son information

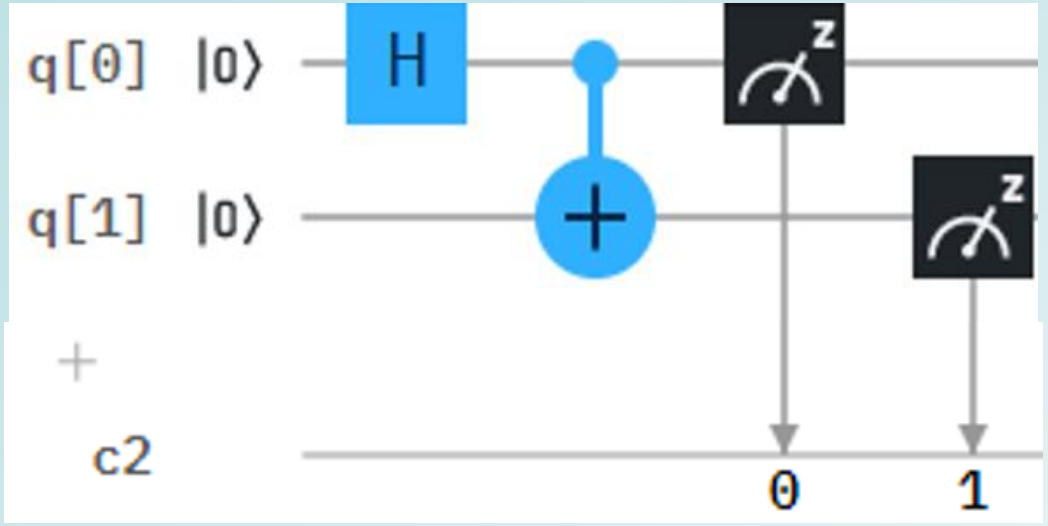
Les portes quantiques

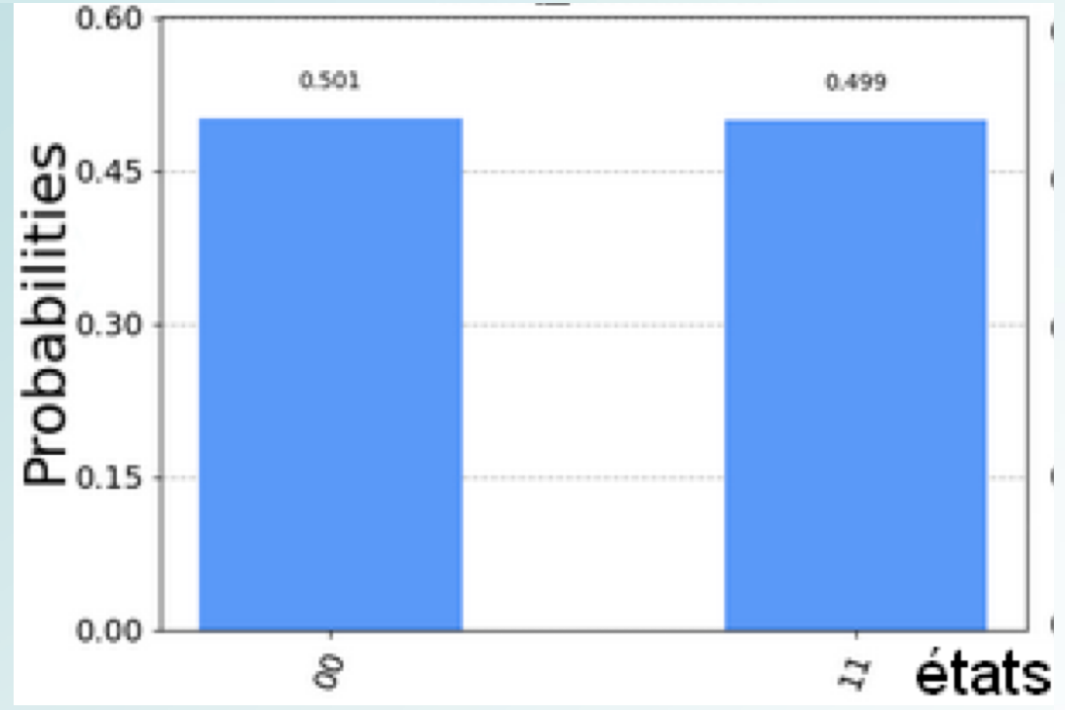
- Il est possible de faire des opérations sur les qubits sans détruire les états quantiques
- Ces opérateurs peuvent être représentés par des portes ou des matrices 2 x 2 unitaires (préservant la norme)
- Nous allons nous intéresser à 4 portes
  - **X** : bitFlip (équivalent au NOT)
  - **H** : permettant de ***superposer*** deux états
  - **CNOT** : porte binaire ***intriquant*** des états
  - **M** : permettant de ***mesurer*** un qubit

- H, S, T et CNOT forment un ensemble de portes universelles

Operator	Gate(s)	Matrix
Pauli-X (X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Hello World (superposé et intriqué)



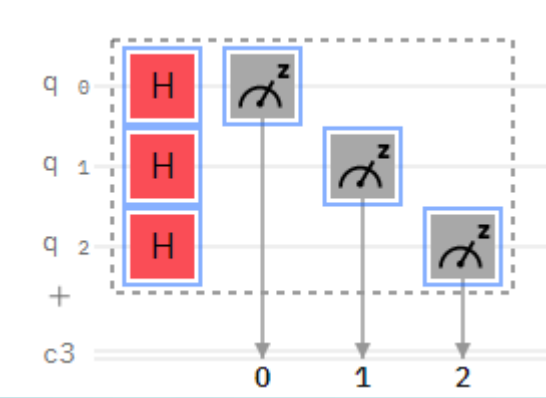


« Etat de Bell »

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

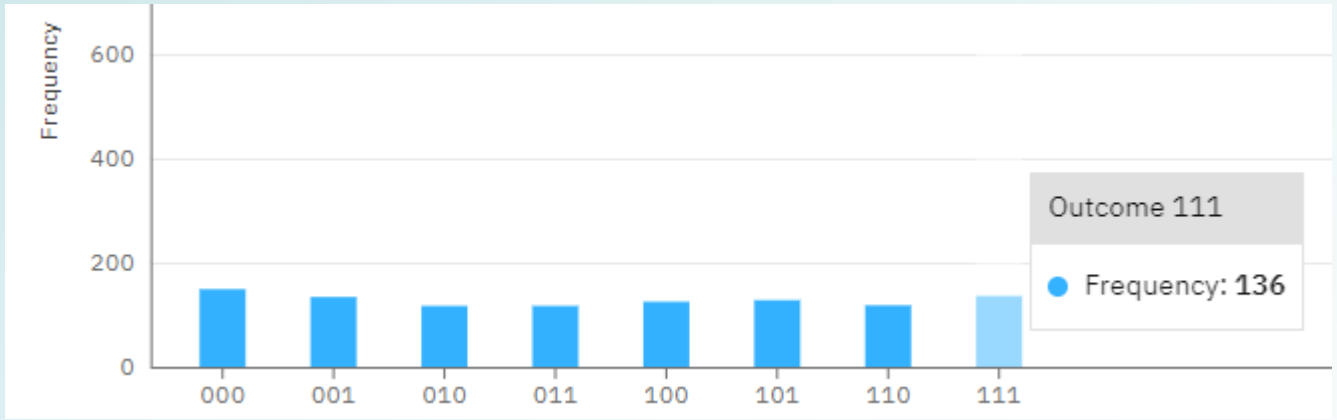
~~$|\varphi_1\rangle \otimes |\varphi_2\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle$~~

Représenter des entiers (simultanément)



- Un registre de 3 QuBits s'écrit $|q_2q_1q_0>$ (représentation binaire d'un entier)
- Chaque Qubit superpose l'état |0> et |1> équitablement (porte H)
- La lecture des 3 QuBits (portes grises) fournit aléatoirement un chiffre entre 0 et 7
- L'exécution 1000 fois de ce circuit fournit une distribution de probabilités

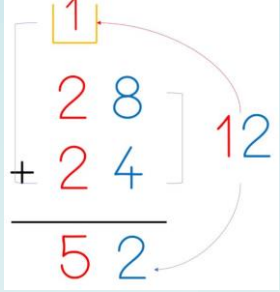
Exemple : 3 qubits, 8 états



"The rule of simulation that I would like to have is that the number of computer elements required to simulate a large physical system is only to be proportional to the space-time volume of the physical system. I don't want to have an explosion. That is, if you say I want to explain this much physics, I can do it exactly and I need a certain-sized computer. If doubling the volume of space and time means I'll need an exponentially larger computer, I consider that against the rules" Feynmann 1982

Remarque : 40 QuBits peuvent « stocker » un Tera d'états.

L'addition (adder)



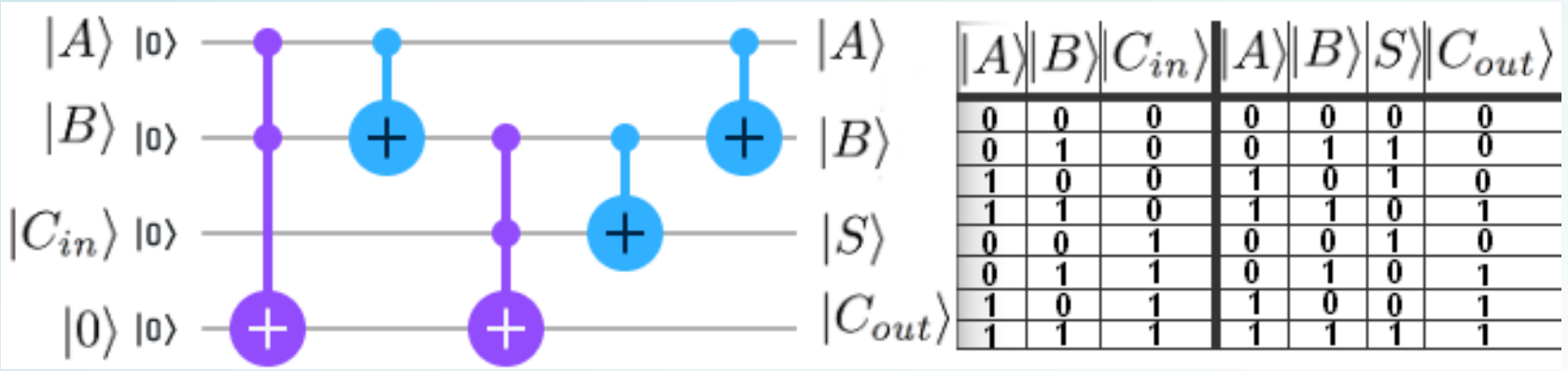
$^{1\,1}1\,1\,1\,0\,0$
$1\,1\,0\,0\,0$

$1\,1\,0\,1\,0\,0$

- Des entier A,B représentés sur des registres de taille $\lceil \log_2(A) \rceil$ et $\lceil \log_2(B) \rceil$

  $A = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_2 2^2 + a_1 2^1 + a_0 2^0$
- Somme entre deux entiers comme à l'école : A+B=S

$$
\begin{array}{cccccccccc}
c_{n-1} & c_n & c_{n-1} & \dots & c_2 & c_1 & c_0 & & & \\
 & b_n & b_{n-1} & \dots & b_2 & b_1 & b_0 & & & \\
+ & a_n & a_{n-1} & \dots & a_2 & a_1 & a_0 & & & \\
\multicolumn{10}{c}{==========} \\
s_{n+1} & s_n & s_{n-1} & \dots & s_2 & s_1 & s_0 & & &
\end{array}
$$

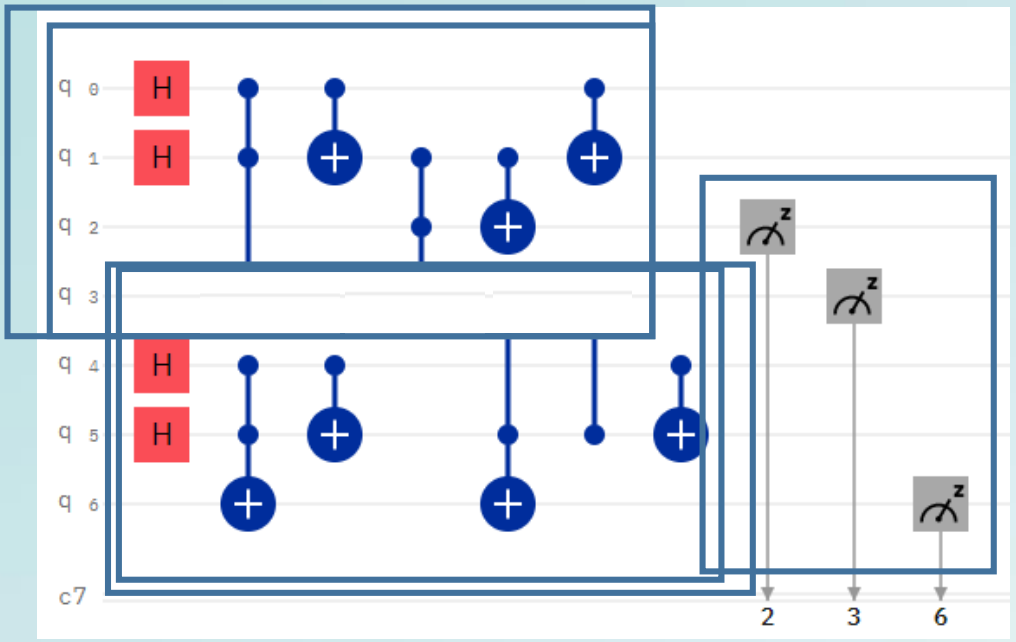


$\|A\rangle$	$\|B\rangle$	$\|C_{in}\rangle$	$\|A\rangle$	$\|B\rangle$	$\|S\rangle$	$\|C_{out}\rangle$
0	0	0	0	0	0	0
0	1	0	0	1	1	0
1	0	0	1	0	1	0
1	1	0	1	1	0	1
0	0	1	0	0	1	0
0	1	1	0	1	0	1
1	0	1	1	0	0	1
1	1	1	1	1	1	1

Correction Somme

Additionnons 2 registres A et B (stockés sur 2 QuBits) dans S (3 QuBits)

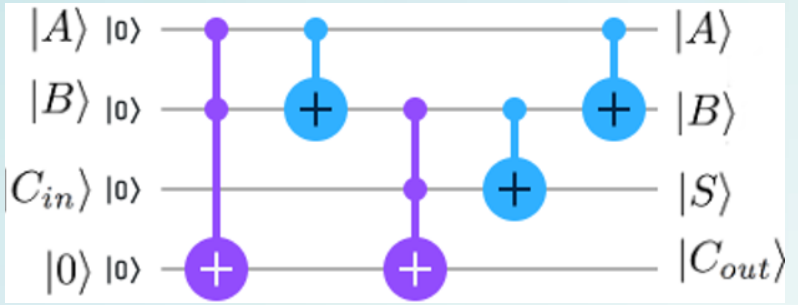
A = |q4q0> B = |q5q1> S=|q6q3q2>

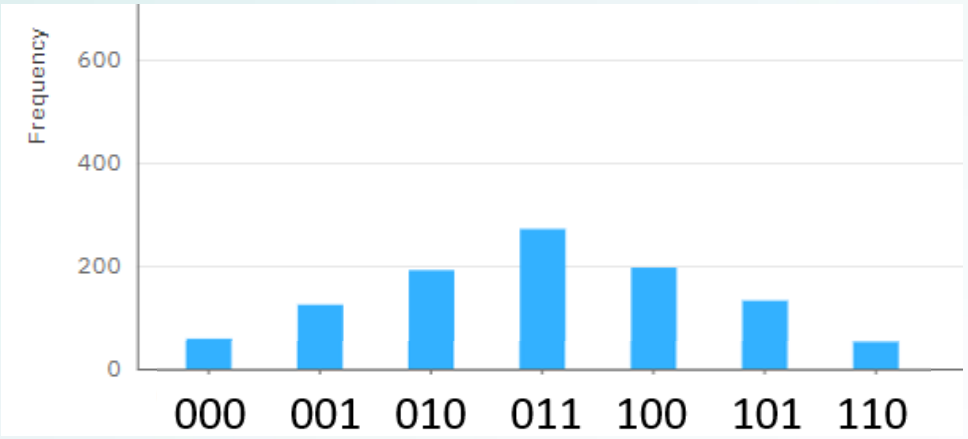


a_0
b_0
0 $..s_0$ ←
$..c_1..s_1$ ←
a_1
b_1
$..c_2$ ←

$c_2\ c_1$
b_1b_0
$+ a_1a_0$
=========
$c_2s_1s_0$

A D D E R

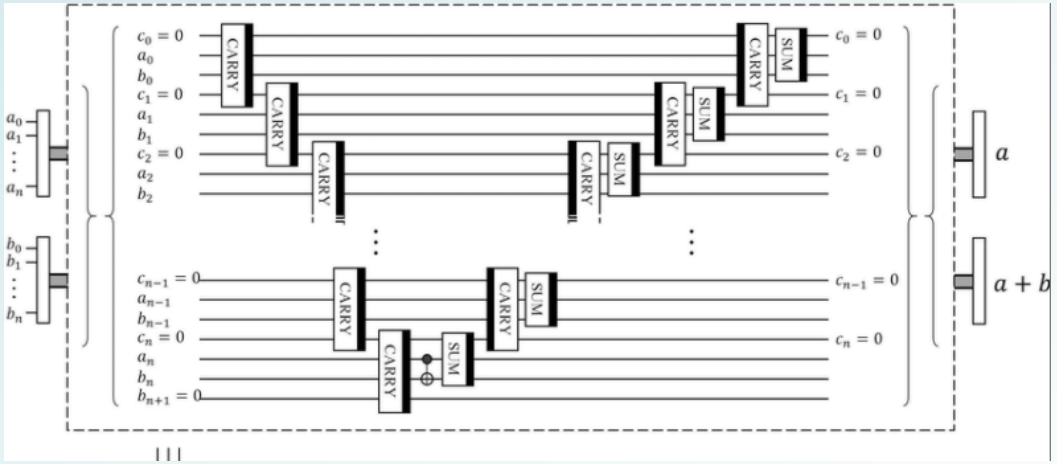




- Il « manque » l'état 111
- Les probabilités correspondent à la convolution des deux lois de probabilité uniforme … en 0(10)

Dérivés de la somme

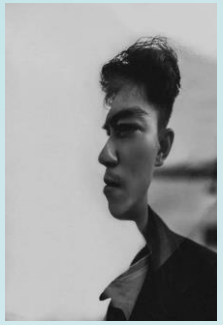
- Circuit itéré de la somme (et bien plus)
  - *« Quantum Networks for Elementary Arithmetic Operations », Vedral et al, Physical Review 95*
- Circuit optimisé de la somme
  - *Steven Cuccaro, Thomas Draper, Samuel Kutin, and David Moulton. A new quantum ripple-carry addition circuit. 11 2004. 15*
  - (a,b) => (a,a+b)+1 carry
- Circuit de la soustraction
  - *Circuit de la somme de droite à gauche*
  - (a,b) => (a,b-a)
- Circuit de la comparaison
  - À partir de la soustraction, on test le résultat du carry : 0 si b≥a et 1 sinon, on inverse le carry (car on cherche le booleen a<b) puis on refait l'addition
  - (a,b) => (a,b-a)+carry=>(a,b)+carry « a < b »



L’informatique quantique

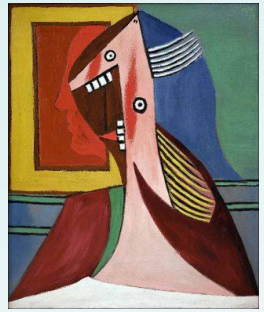
Bit d’info de profil

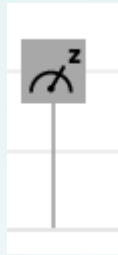
0 1



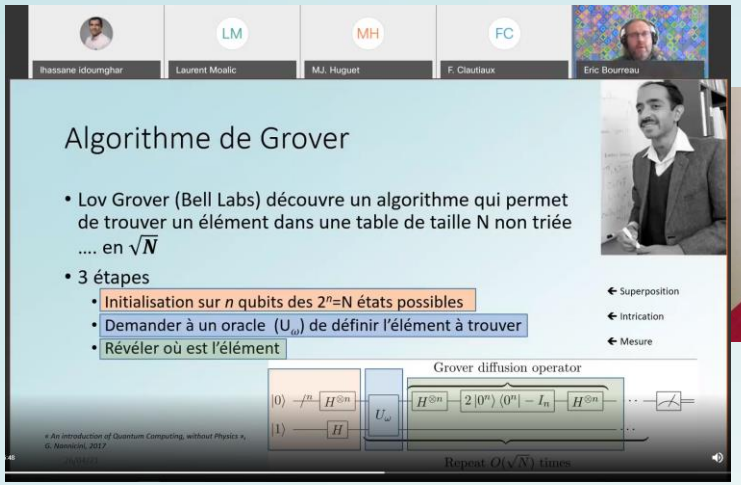
Superposition

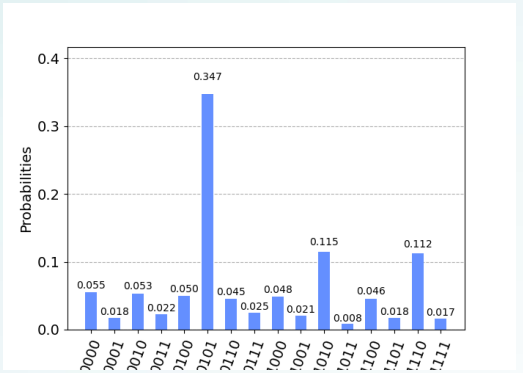
Déformation de la fonction d’onde

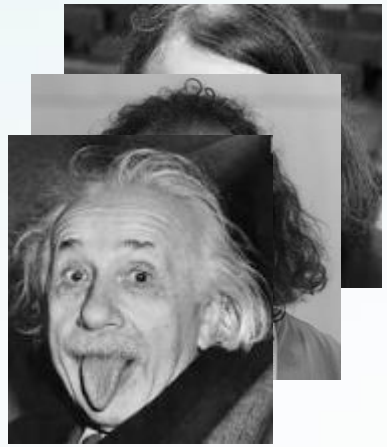




Mesure







https://roadef2021.sciencesconf.org/resource/page/id/11

Définition d’une Recherche Opérationelle Quantique (26/04/2021)

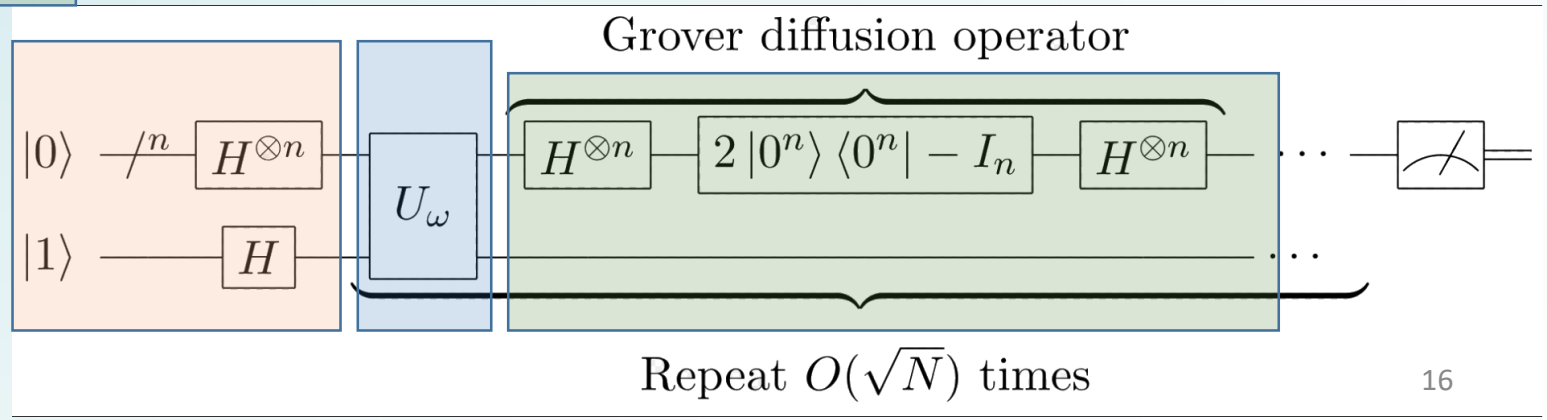
Algorithme de Grover

- Lov Grover (Bell Labs) découvre un algorithme qui permet de trouver un élément dans une table de taille N non triée …. en $\sqrt{N}$
- 3 étapes
  - Initialisation sur *n* qubits des 2^n=N états possibles
  - Demander à un oracle (U_ω) de définir l'élément à trouver
  - Révéler où est l'élément

← Superposition

← Intrication

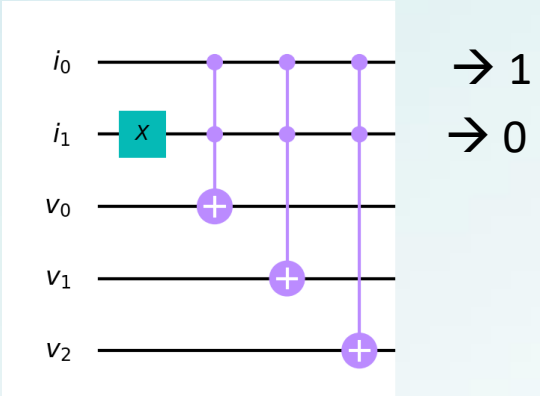
← Mesure



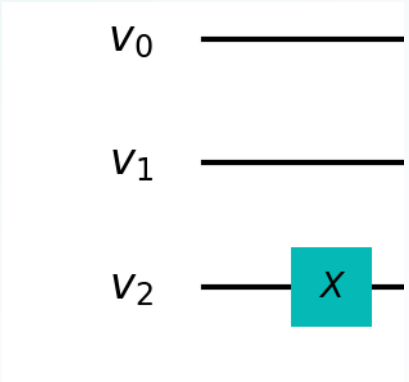
« An introduction of Quantum Computing, without Physics », G. Nannicini, 2017

Accesseur dans un tableau

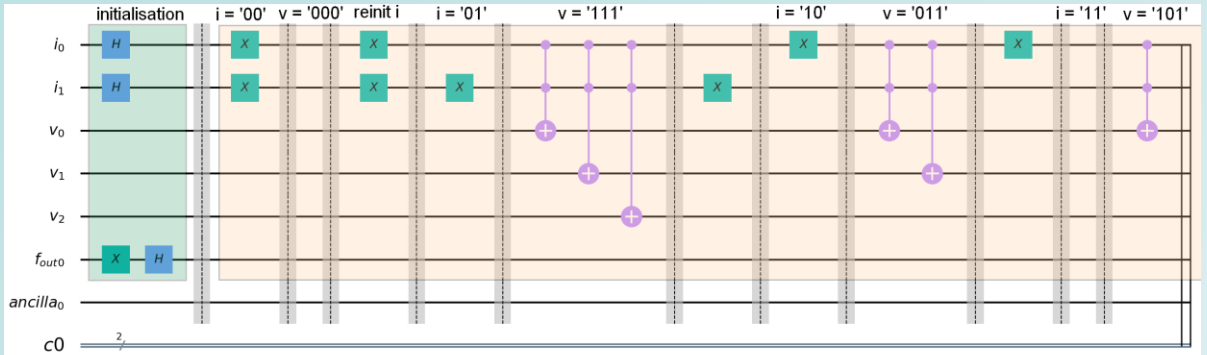
- T = [0,7,3,5] on cherche l'élément e=3 dans le tableau. On veut déduire son indice.
- Soient 2 qubits pour les indices (allant de 0 à 3) superposés
- Soient 3 qubits pour les valeurs de T (allant de 0 à 7)
- Encoder 7 à la position 1, c'est encoder les valeurs v_i avec 1,1,1 et les indices i_1,i_0 avec 0,1

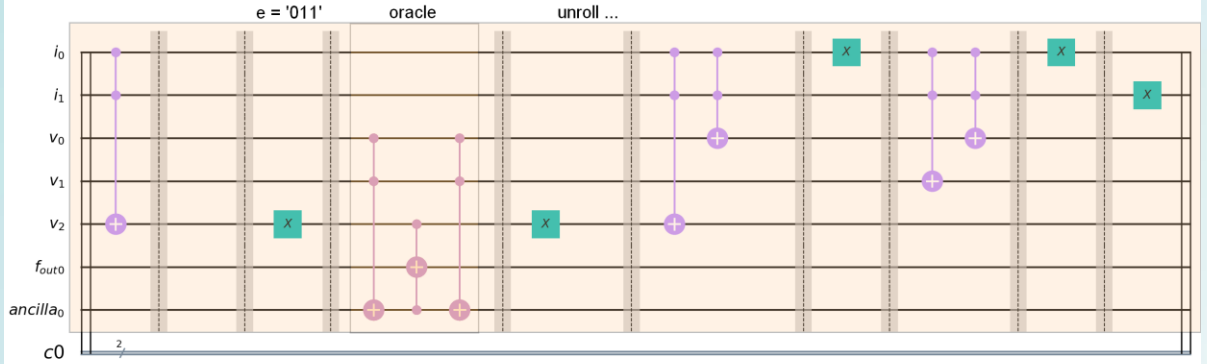


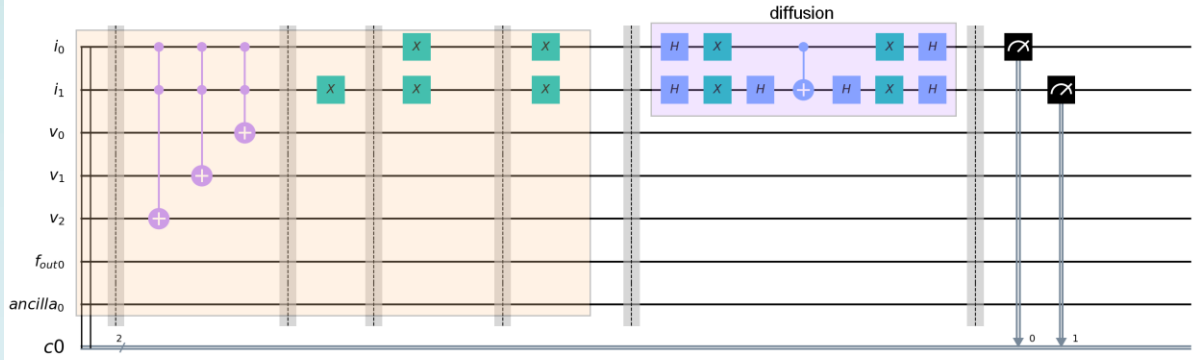
rechercher e=3, c'est encoder 001 sur v_i

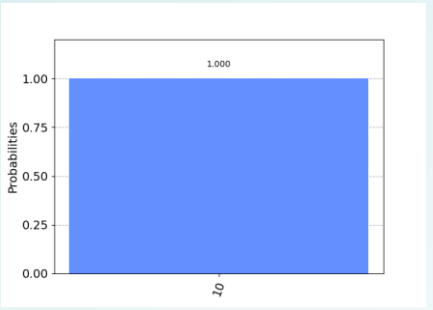


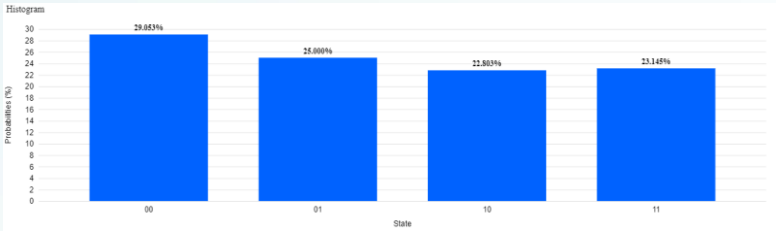
Accesseur dans un tableau - T = [0,7,3,5] e=3











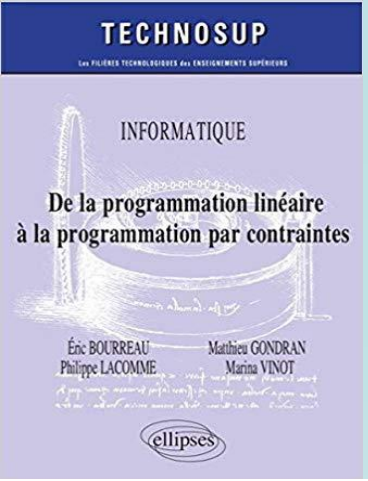
Optimiser, c’est décider

- Recherche d’un minimum avec Grover
- « Existe-t-il une solution < k ? », si oui itérer

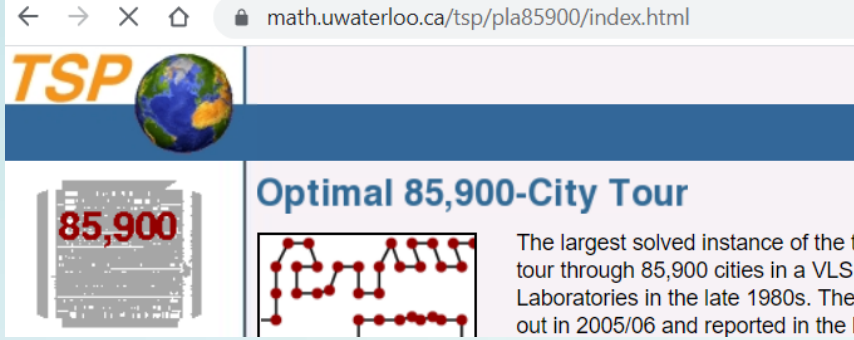
A quantum algorithm for finding the element, C. Durr, 1996

- Si *M* solutions possibles, complexité de Grover $O(\sqrt{\frac{N}{M}})$
- Si nombre de solutions inconnu

```
Iter_max = 1
Tant que iter < sqrt(N)
|     iter_max = iter_max * 4/3
|     iter = random(iter_max)
|     Grover avec iter itérations  ➔ si solution trouvée : goto fin
FinTantQue
Fin
```



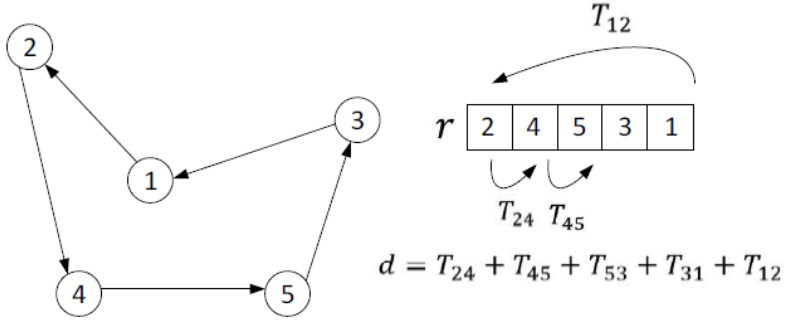
Voyageur de commerce



5.9 Modélisation PPC du TSP

5.9.1 *Principe général*

Une solution du TSP peut s'écrire sous la forme d'un vecteur r : par exemple, r = [2; 4; 5; 3,1] représente la tournée : 2-4-5-3-1 comme le montre la Figure 5-28.



$$d = T_{24} + T_{45} + T_{53} + T_{31} + T_{12}$$

Figure 5-28.Une solution modélisée avec le vecteur r

Le coût d'une solution (distance totale) est la somme des distances à parcourir entre deux villes successives. Pour la tournée précédente, la distance est $d = T_{2,4} + T_{4,5} + T_{5,3} + T_{3,1} + T_{1,2}$.

Modélisation PPC du problème

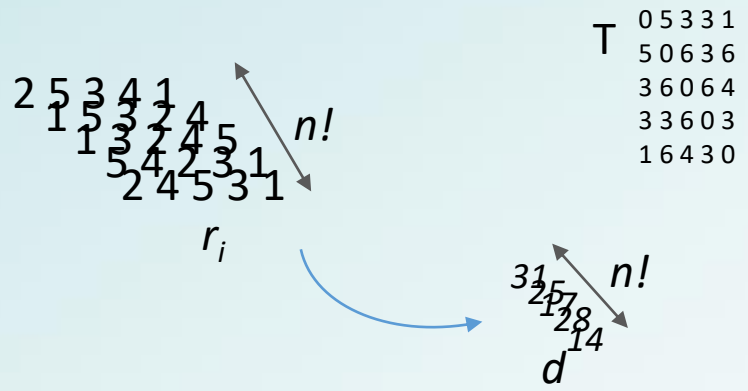
$\forall i = 1..N \qquad r_i \in [1; n]$

$\forall i = 1..N, \forall j = 1..N \qquad r_i \neq r_j$

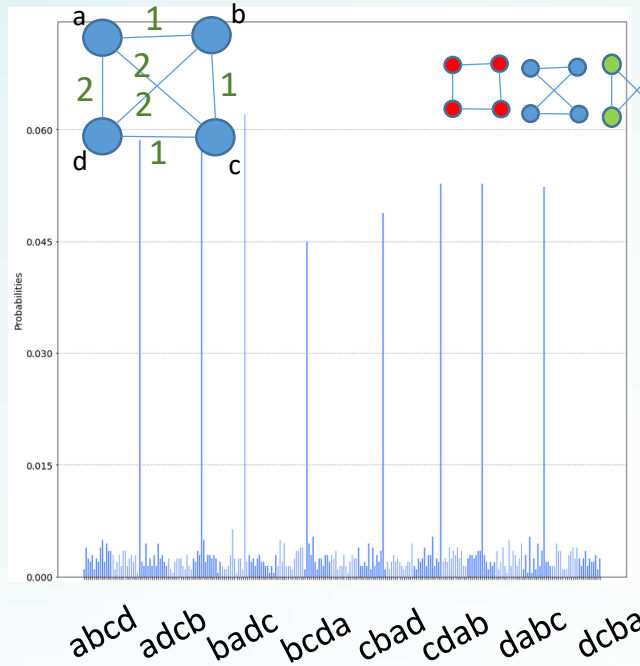
$$d = \sum_{i=1}^{N-1} T_{r_i, r_{i+1}} + T_{r_N, r_1}$$

$Min\ d$

- Le problème consiste à visiter une et une seule fois chaque ville en minimisant la distance totale parcourue.
- Modélisation Quantique

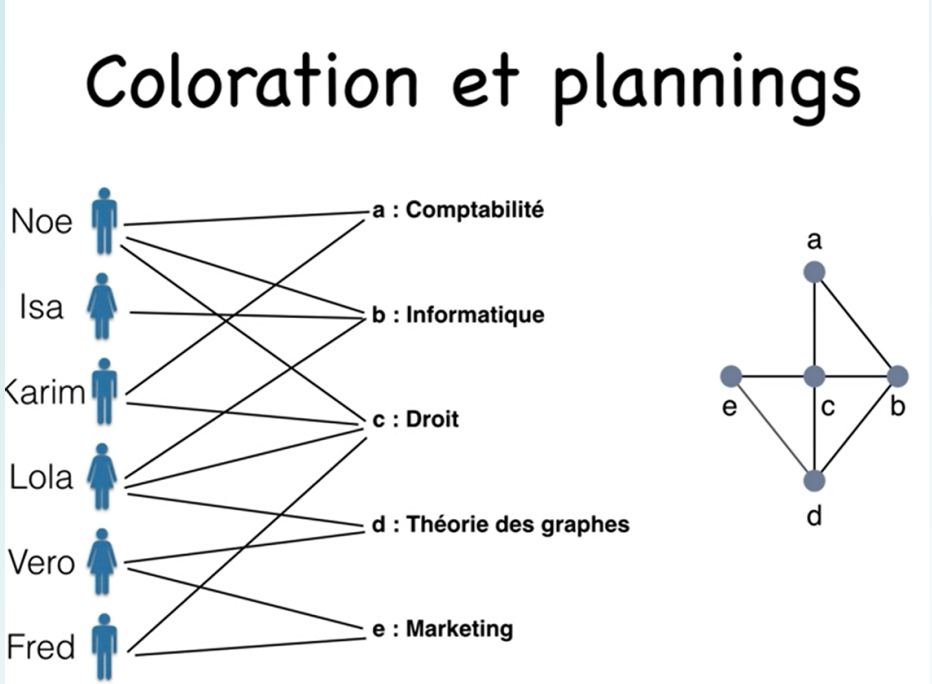


- Minimiser somme sur i des $\boldsymbol{D_{P_i P_{i+1}}}$
  ➔ accesseur d'une table
  $(i, P_i, P_{i+1}, \boldsymbol{D_{P_i P_{i+1}}})$
- Utiliser Grover pour exhiber le minimum de *d*



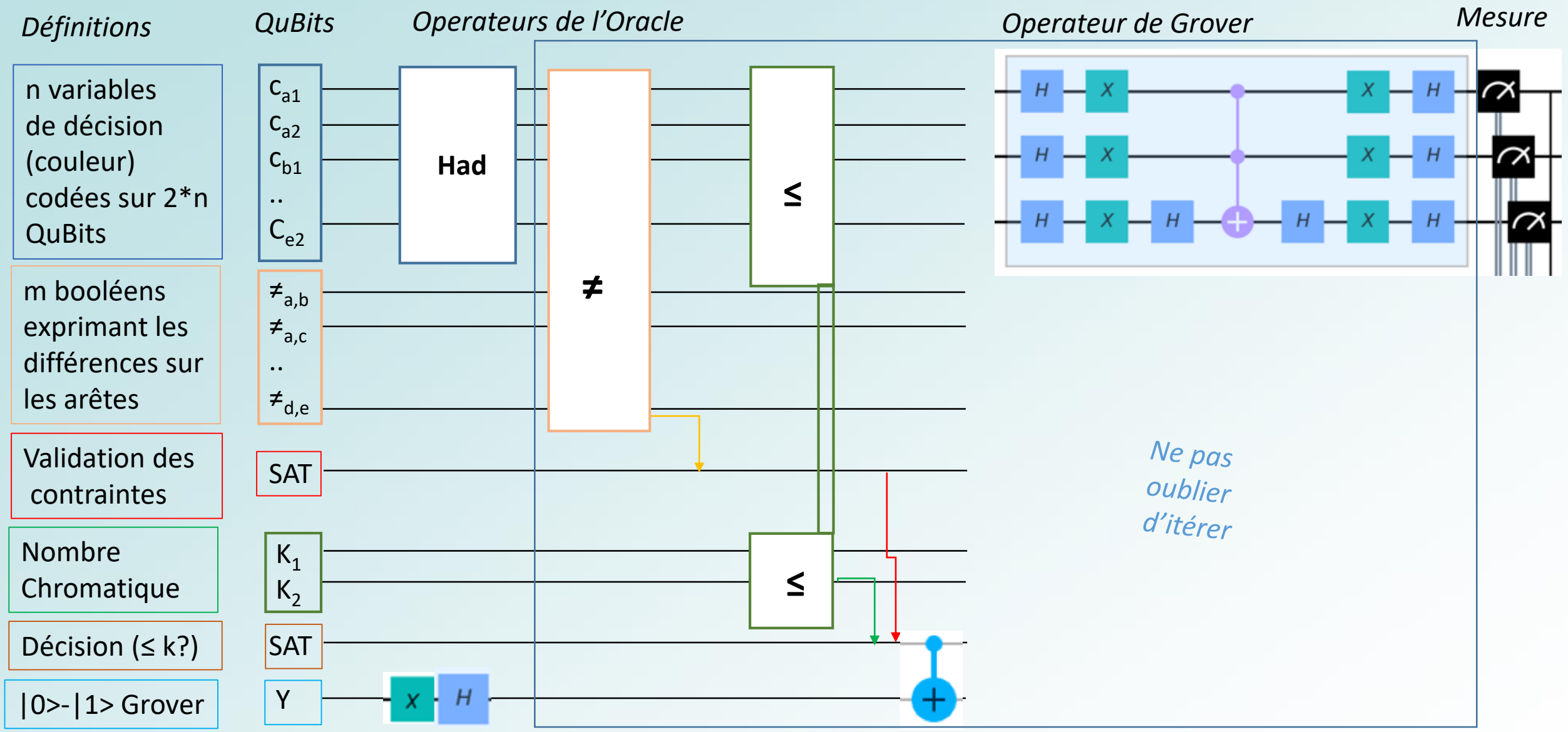
GT R novembre 2022 - TP Grover, Exercice 3 Encoder un problème de coloration

- Représenter un problème sous forme de graphe
  - Construction d'un emploi du temps
  - Un coloration du graphe des conflits fourni un planning
- Superposition
  - Variables de décision (couleur sur 2 QuBits)
  - H^{2n}
- Oracle (intrication)
  - Contraintes à satisfaire (opérateur 'différent')
  - Fonction de coût à minimiser (opérateur '≤')
- Opérateur de Grover (révélateur)
- Itérer
- Mesurer



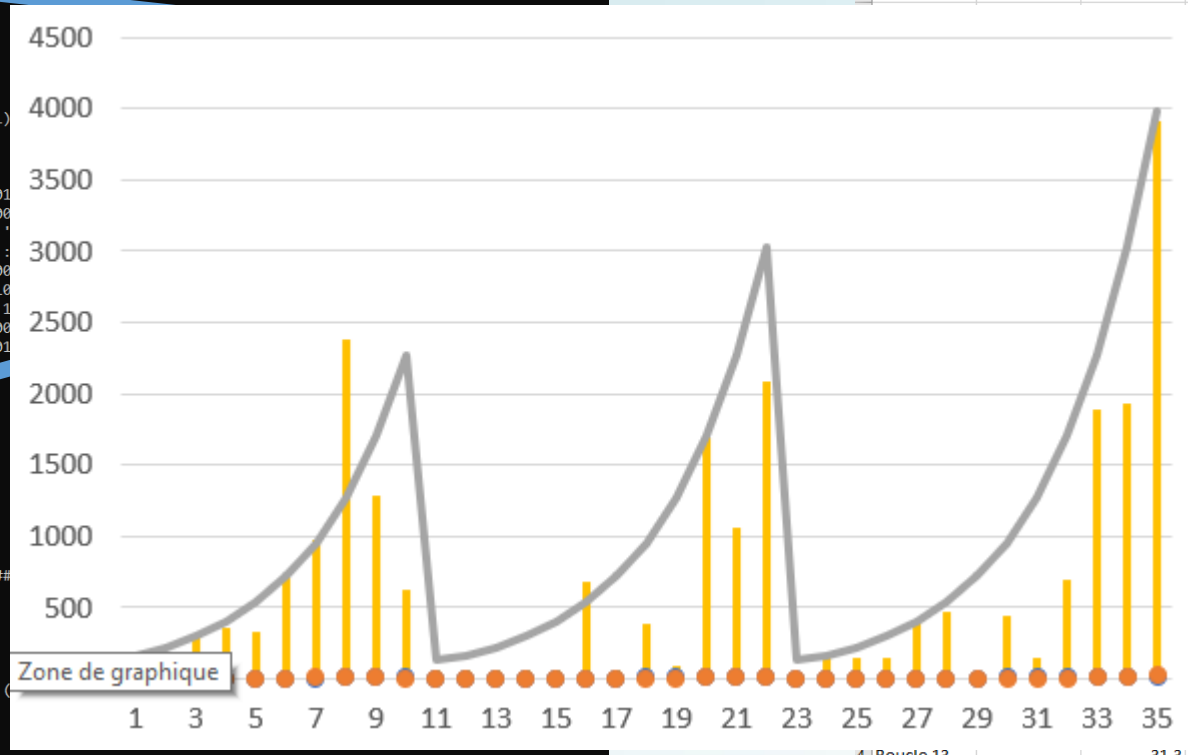
https://www.youtube.com/watch?v=CUe7LC3CdH8&ab_channel=%C3%80lad%C3%A9couvertedesgraphes

Circuit Quantique pour la coloration avec Grover



Exercice 3
Exemple d'exécution

```
C:\WINDOWS\system32\cmd.exe - python  color.py
{ '1111111111': 1, '1110011000': 1, '1101111101': 1, '1110001011': 1, '0000110011': 1, '1000011001': 1, '0011100010': 1, '0111111010': 1, '1011010111'
1': 1, '1010001111': 1, '1100010100': 1, '1110001111': 1, '1111000111': 1, '1111111010': 1, '1001110010': 1, '0111000001': 1, '0000000001': 1, '111100
100000': 1, '1011001100': 1, '1001001100': 1, '0010001010': 1, '0101110000': 1, '0111010101': 1, '0110101110': 1, '1110000101': 1, '1100000111': 1}
Etat le plus probable : 0000110010 nb = 2
Sommet  0 de couleur 0
Sommet  1 de couleur 0
Sommet  2 de couleur 3
Sommet  3 de couleur 0
Sommet  4 de couleur 1
Arete violee  0 , 1
Arete violee  1 , 3
Nb co =  7
    Solution non valide
    Recherche : 17.757726633812588/32.0
        Nombre d'itérations : 5
        Circuit composé de : 35 qubits, 10 bits.
        Portes :  OrderedDict([('cx', 690), ('ccx', 480), ('x', 471), ('h', 111)
\********** Execution du circuit **********/
Circuit executé en 624.9214503765106 s.
\********** Lecture des états **********/
{'1110110101': 1, '1100110110': 1, '0011011111': 1, '0101011111': 1, '1100011101
': 1, '0110101010': 1, '1001101110': 1, '1110101110': 1, '0011111110': 1, '10000
10001': 1, '0101101111': 1, '0100100010': 1, '0001110010': 1, '0100001101': 1, '
1010111001': 1, '1010101011': 1, '0110011010': 1, '0011011011': 1, '1011010011':
 1, '1110010110': 1, '1110000010': 1, '0110111111': 1, '1011100100': 1, '1010000
010': 1, '1110100101': 1, '1101100000': 1, '1010100000': 1, '0010000011': 1, '10
10010000': 1, '1000000001': 1, '1111110111': 1, '0111010011': 1, '1010011111': 1
 '1110001010': 1, '0000000111': 1, '1101001111': 1, '1101010001': 1, '0101110000
0': 1, '0000111010': 1, '0000101000': 1, '0010111111': 1, '0101110010': 1, '0101
000100': 1, '1010010011': 1, '1111000100': 1, '1000011111': 1, '1110001100': 1,
'0101000001': 1}
Etat le plus probable : 1100101101 nb = 2
Sommet  0 de couleur 3
Sommet  1 de couleur 0
Sommet  2 de couleur 1
Sommet  3 de couleur 3
Sommet  4 de couleur 2
Nb co =  3
    Solution valide
Etat le plus probable : 1100101101 nb = 2
################################################################################
Fenêtre de recherche : 1 - 4
Recherche d'une solution de poids <= 2
    Recherche : 1/32.0
        Nombre d'itérations : 1
        Circuit composé de : 35 qubits, 10 bits.
        Portes :  OrderedDict([('cx', 138), ('ccx', 96), ('x', 95), ('h', 31), ('
\********** Execution du circuit **********/
Circuit executé en 64.42988657951355 s.
\********** Lecture des états **********/
{'0111000110': 1, '0100100001': 1, '1000111010': 1, '1001010011': 1, '0000000110': 1, '1000111100': 1, '1001010001': 1, '1101000011': 1, '0100011111':
': 1, '0001000110': 1, '0001100011': 1, '1101101111': 1, '0110011110': 1, '0010111110': 1, '1011110100': 1, '1000000000': 1, '1100001001': 1, '0001111
11000': 1, '0010001001': 1, '0000011011': 1, '0100111100': 1, '0010111001': 1, '1011101011': 1, '1110100111': 1, '0001010111': 1, '1100100111': 1, '01
0110101000': 1, '0011101100': 1, '0100101001': 1, '1000000111': 1, '1100100011': 1, '1110111010': 1, '0011000011': 1, '1001100110': 1, '1000001010': 1
 1, '0110110111': 1, '0011100000': 1, '0011001011': 1, '0001000011': 1, '1010000010': 1, '1010011010': 1, '0111111100': 1, '1001110001': 1, '100100110
011': 1, '0100100101': 1, '1010101100': 1, '0110000100': 2, '1001001110': 1, '0000000011': 1, '1011000100': 1, '1101001110': 1, '0111011100': 1, '0010
```



4 couleurs ?								
circuit 35 qubits dont 10 mesurés								
1024 états			Taille du circuit					
	iter	iter_max	CX	CCC	X	H	M	
	temps (s)	solution	nb viols	nb couleurs	nouvelle solution améliorante ?			
Boucle 1	1	1	138	96	95	31	10	
	63	1-2-3-2-1	1	3	Non		63	
Boucle 2	2	1,3	276	198	189	51	10	
	74	3-1-1-3-3	2	2	Non		137	
Boucle 3	2	1,7	276	198	189	51	10	
	66	1-3-1-2-0	1	4	Non		203	
Boucle 4	3	2,3	414	288	283	71	10	
	313	0-3-1-1-1	3	3	Non		516	
Boucle 5	4	3,1	552	384	377	91	10	
	360	3-3-1-1-1	4	2	Non		876	
Boucle 6	3	4,2	414	288	283	71	10	
	326	1-1-1-0-3	3	3	Non		1202	
Boucle 7	5	5,6	690	480	471	111	10	
	705	1-1-3-2-0	1	4	Non		1907	
Boucle 8	8	7,4	1104	768	753	171	10	
	983	1-1-2-0-3	1	4	Non		2890	
			1242	864	847	191	10	
			1	3	Non		5276	
			1242	864	847	191	10	
			2	3	Non		6561	
			690	480	471	111	10	
			0	4	Oui		7185	
			1	3	Non		7249	
			1	4	Non			
			1	4	Non			
			1	3	Non			
			0	4	Non			
			3	2	Non			
			2	3	Non			
			2	3	Non			
			1	3	Non			
			1656	1152	1129	251	10	
			2	3	Non			
			2	3	Non			
			2760	1920	1881	411	10	
Boucle 13		31,3						
	?							
PREUVE								

Conclusion et perspectives

- Exemples et codes disponibles sur le git du LIRMM
  https://gite.lirmm.fr/bourreau/quantumgroversearch
  (somme, table, SAT, vertex cover, tsp, coloring)
- De nouveaux algorithmes à inventer
  - Compromis entre la combinatoire et les propriétés quantiques des machines
  - Arbre de backtrack superposé sur une descente
    + estimation probabiliste = réduction quadratique de l'exploration

    Quantum Walk speedup of backtracking algorithms, Montanaro 2016

    Quanum algorithm for tree size estimation, Ambainis 2017
  - Algorithme de filtrage au sein de contraintes globales
  - Accélérateur de points intérieurs pour le SDP
  - …