# Grover

E. Jeandel

Université de Lorraine, France

# Plan

# Plan

# Problématique

Algorithmes avec oracle (boîte noire)

$f : \{0,1\}^n \to \{0,1\}$ une fonction, donnée par un oracle $U_f$.

Rappels

- $U_f \left|x\right\rangle = (-1)^{f(x)} \left|x\right\rangle$
- $V_f \left|xy\right\rangle = \left|x\right\rangle \left|f(x)+y\right\rangle$

Complexité mesurée en nombre d'appels à l'oracle + temps de calcul.

# Problématiques typiques

- Trouver $x$ tel que $f(x) = 1$.
  - Algorithme de Grover
- Trouver $p$ tel que $\forall x, f(x + p) = f(x)$
  - Algorithme de Shor (+ = Addition) ou de Simon (+ = xor)
- Trouver $s$ tel que $\forall x, f(x) = x \cdot s$
  - Bernstein-Vazirani

# Algorithme de Grover - Énoncé imprécis de l'algorithme

Soit $f : \{0,1\}^n \to \{0,1\}$ une fonction. On note $N = 2^n$.

L'algorithme de Grover (1996) trouve $x$ tel que $f(x) = 1$ en temps $O(\sqrt{N})$.

# Énoncé un peu plus précis de l'algorithme

Soit $f : \{0,1\}^n \to \{0,1\}$ une fonction. On note $N = 2^n$.

L'algorithme de Grover trouve $x$ tel que $f(x) = 1$ en temps $O(\sqrt{N})$ et avec $O(\sqrt{N})$ appels à $U_f$.

Rappel: $U_f \left|x\right\rangle = (-1)^{f(x)} \left|x\right\rangle$

# Comparaison classique-quantique

En classique:

- De façon déterministe, on doit tester toutes les possibilités. Le meilleur algo fait donc $N$ requêtes à la fonction $f$.
- En probabiliste, si on teste les $x$ dans un ordre aléatoire la complexité moyenne est $N/2$ (si une seule solution)

L'algorithme quantique est quadratiquement meilleur!

- Bennett-Bernstein-Brassard-Vazirani (1997): $\sqrt{N}$ est optimal.
  - Le gain n'est QUE quadratique.

# Plan

# Arnaques

Attention aux arnaques !

Soit $T$ un tableau de taille $N$, Je cherche un 1 dans le tableau.

Est-ce que l'algorithme de Grover prend un temps $O(\sqrt{N})$ ?

# Arnaques

Soit $T$ un tableau de taille $N$, Je cherche un 1 dans le tableau.
Est-ce que l'algorithme de Grover prend un temps $O(\sqrt{N})$ ?

Non

Il faut (a) calculer $U_T$ (b) faire $\sqrt{N}$ appels à $U_T$.

- Bien malin qui calcule $U_T$ sans lire tout le tableau!
- En règle générale, si $T$ est générique, $U_T$ sera de profondeur $O(\log N)$ voire $O(N)$ suivant l'architecture matérielle.

L'algorithme de Grover sera donc en $O(N\sqrt{N})$ sur un tableau quelconque !
(Aux termes logarithmiques près)

# Intérêt

Grover est donc intéressant:

- Si l'oracle $U_f$ nous est offert (on ne compte pas sa complexité)
  - QRAM ??
- Si $f$ est facile à calculer (donc $U_f$ est de petite taille)

# Exemples

## Problèmes NP

- Soit $\phi$ une formule 3CNF, trouver $S$ tel que $\phi[S] = 1$.
  - Grover en $\sqrt{2^n} = 2^{n/2} = 1.414^n$.
  - Meilleur algo classique en $1.308^n$.
- Trouver une 3-coloration d'un graphe
  - Grover en $\sqrt{3^n} = 1.732^n$ naivement
  - Meilleur algo classique en $1.329^n$
- Etant donné $x, y$, trouver $K$ tel que $DES(K, x) = y$
  - Applications en crypto

# Plan

# Monte Carlo

L'algorithme de Grover est un algo de Monte-Carlo: Il trouve une solution, si elle existe, avec probabilité $1 - \frac{1}{N}$.

- L'algorithme de Grover peut échouer, mais la probabilité est très faible
  - On peut toujours le répéter pour la diminuer
- Il faut en tenir compte dans certaines applications
  - Ex: On cherche pour tous les $x$, un $y$ tel que $f(x, y) = 1$.
  - La complexité totale n'est pas $N\sqrt{N}$ mais $N\sqrt{N}\log N$.
  - Il faut répéter pour être certain de trouver.
  - En règle générale, des facteurs $\log N$ arrivent dans toutes les applications naives.
- Si la fonction est constante égale à 0, on ne peut pas le savoir.
  - Ne résout pas des problèmes de décision au sens NP.

# Uniformité

L'algorithme de Grover renvoie chaque solution avec la même probabilité: on a donc une distribution uniforme sur toutes les solutions de $f(x) = 1$.

- Important pour les applications, surtout quand il faut le répéter.

# Plan

# Principe de l'algorithme de Grover

## Etape 1

Produire une superposition de toutes les entrées au problème.

Si on mesure, tous les $x$ sont équiprobables, ce n'est pas très utile.

## Etape 2

Modifier l'état du système de façon à augmenter les amplitudes des $x$ tels que $f(x) = 1$.

Si on mesure, on a plus de chance de mesurer un $x$ tel que $f(x) = 1$

## Etape 3

Mesurer

Profit.

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.0625 = 1/16

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.47

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.90

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.96

# Attention

**Attention**

La phase d'amplification applique un circuit quantique, elle ne peut PAS mesurer.

**Attention**

# Principe de l'algorithme de Grover

## Etape 1

Produire une superposition de toutes les entrées au problème.

Comment faire l'étape 1 ?

## Etape 2

Modifier l'état du système de façon à augmenter les amplitudes des $x$ tels que $f(x) = 1$.

## Etape 3

Mesurer

# Etape 1

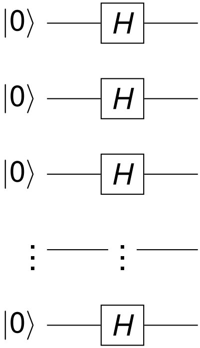
**Comment faire l'étape 1 ?**

Partant de l'état $|000\ldots0\rangle$, il suffit d'appliquer l'opérateur d'Hadamard $H$ sur tous les qubits, pour obtenir:

$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle$$

# Etape 1

Comment faire l'étape 1 ?



# Principe de l'algorithme de Grover

## Etape 1

Produire une superposition de toutes les entrées au problème.

## Etape 2

Modifier l'état du système de façon à augmenter les amplitudes des $x$ tels que $f(x) = 1$.

Comment faire l'étape 2 ?

## Etape 3

Mesurer

# Un petit théorème au passage

**Theorem**

*Soit $U$ une matrice unitaire (un circuit quantique). Alors il existe $n$ tel que $U^n \simeq I$.*

Si on applique le même circuit suffisament longtemps, on revient sur la configuration initiale

A un moment donné, les amplitudes vont redescendre!!

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.0625 = 1/16

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.47

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.90

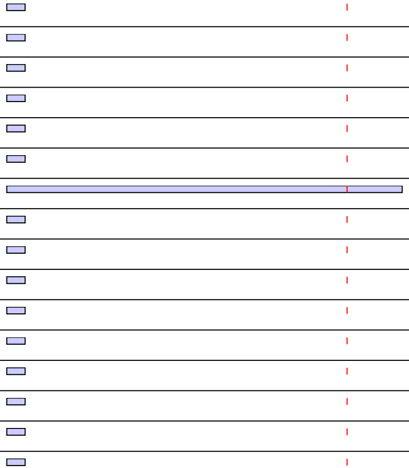
# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.96

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |



Proba de succès: 0.58

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.12

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.02

# Exemple

| x | f(x) | amplitude (au carré) |
|---|---|---|
| 0000 | 0 | |
| 0001 | 0 | |
| 0010 | 0 | |
| 0011 | 0 | |
| 0100 | 0 | |
| 0101 | 0 | |
| 0110 | 1 | |
| 0111 | 0 | |
| 1000 | 0 | |
| 1001 | 0 | |
| 1010 | 0 | |
| 1011 | 0 | |
| 1100 | 0 | |
| 1101 | 0 | |
| 1110 | 0 | |
| 1111 | 0 | |

Proba de succès: 0.36

# Conclusion

Quelque soit l'opérateur qu'on prend pour augmenter les amplitudes, il faudra surtout faire attention à l'appliquer le bon nombre de fois.

Quel peut être cet opérateur ?

## Quel peut être cet opérateur ?

Le seul vecteur non trivial que nous connaissons c'est $|s\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$

- La seule chose qu'on peut faire, c'est une rotation autour de $|s\rangle$, ou une réflection

# Matrice de Householder

Soit $|v\rangle$ un vecteur, la matrice de Householder associée est

$$H_v = I - 2\frac{|v\rangle \langle v|}{\langle v|v\rangle} = I - 2\frac{|v\rangle \, |v\rangle^\star}{\|v\|^2}$$

Il s'agit de la matrice de reflection par rapport à l'hyperplan orthogonal à $v$.

- A quoi ça sert ?

# Contexte

Supposons avoir accès à :

- $U_f \left|x\right\rangle = (-1)^{f(x)} \left|x\right\rangle$
- $H_s$, reflection orthogonale à $\left|s\right\rangle = \frac{1}{\sqrt{2^n}} \sum_x \left|x\right\rangle$

Que peut-on faire avec ?

Notons $|\omega\rangle$ la superposition des solutions (ce qu'on cherche!) et $s'$ la superposition des non solutions:

$$|\omega\rangle = \frac{1}{\sqrt{K}} \sum_{x|f(x)=1} |x\rangle$$

$$|s'\rangle = \frac{1}{\sqrt{N-K}} \sum_{x|f(x)\neq 1} |x\rangle$$

où $K$ est le nombre de solutions.
Ces deux vecteurs sont orthogonaux.

# Notations

- La superposition initiale $|s\rangle$ est dans l'espace engendré par $|\omega\rangle$ et $|s'\rangle$
- Plus exactement:

$$|s\rangle = \sqrt{\frac{K}{N}}\,|\omega\rangle + \sqrt{\frac{N-K}{N}}\,|s'\rangle$$

# Géométrie

Dans l'espace engendré par $|\omega\rangle$ et $|s'\rangle$, $-H_s$ est une réflexion d'axe $|s\rangle$

# Géométrie

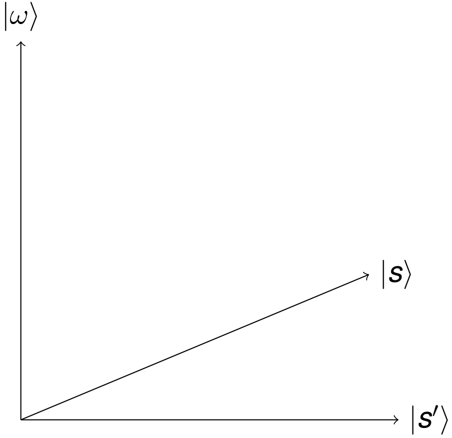
Dans l'espace engendré par $|\omega\rangle$ et $|s'\rangle$, $U_f$ est une réflexion d'axe $|s'\rangle$

Autrement dit $U_f = H_{s'}$

$$\left\{ \begin{array}{rcl} U_f\,|\omega\rangle & = & -\,|\omega\rangle \\ U_f\,|s'\rangle & = & |s'\rangle \end{array} \right.$$

# Dessin

Dans l'espace $|\omega\rangle$ , $|s'\rangle$:



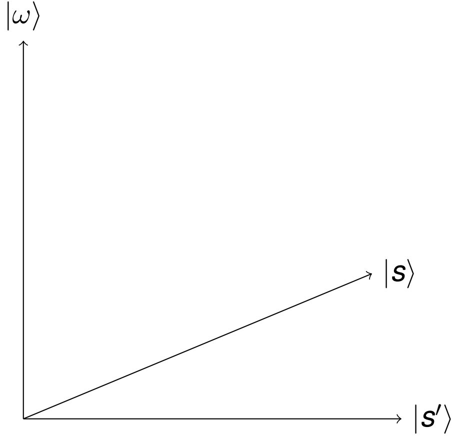

On part de $|s\rangle$, on cherche à obtenir $|\omega\rangle$

- On sait faire une réflexion d'axe à $|s\rangle$
- On sait faire une réflexion d'axe $|s'\rangle$

Que peut-on en faire ?

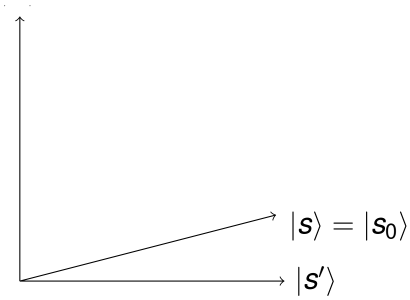
# Dessin

Dans l'espace $|\omega\rangle$ , $|s'\rangle$:



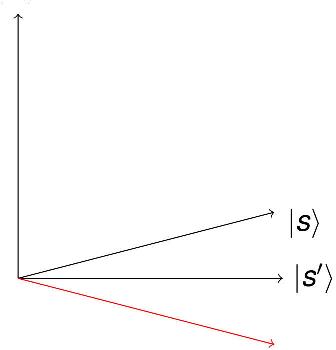

Si on fait d'abord une réflexion d'axe $|s'\rangle$, puis d'axe $|s\rangle$, on se rapproche de $|\omega\rangle$!

# Dessin



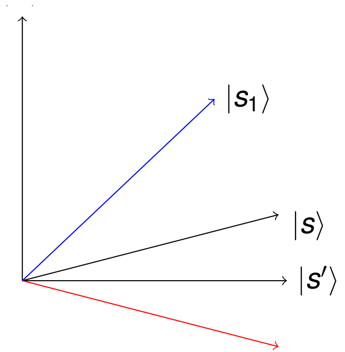

On part de $|s\rangle$, la superposition uniforme de tous les $|x\rangle$.

# Dessin



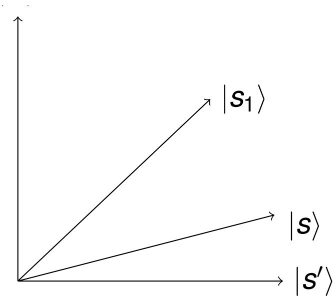

On applique $U_f$, qui effectue la symétrie par rapport à $|s'\rangle$

# Dessin



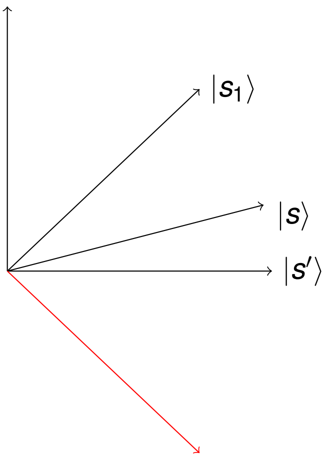

On applique $-H_s$, qui effectue la symmétrie par rapport à $|s\rangle$

# Dessin



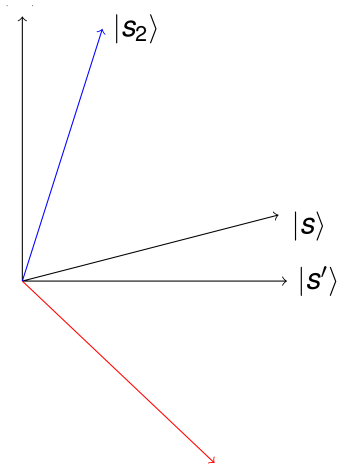

On obtient $|s_1\rangle$. $|s_1\rangle$ est plus proche de $|\omega\rangle$ , donc si on mesure $|s_1\rangle$, on a augmenté la probabilité d'observer un $x$ tel que $f(x) = 1$.

# Dessin



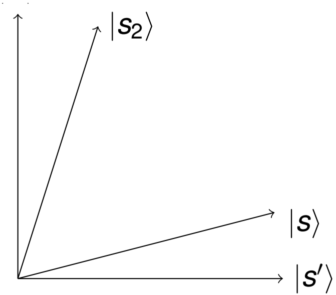

On applique $U_f$ sur $|s_1\rangle$, qui effectue la symmétrie par rapport à $|s'\rangle$

# Dessin



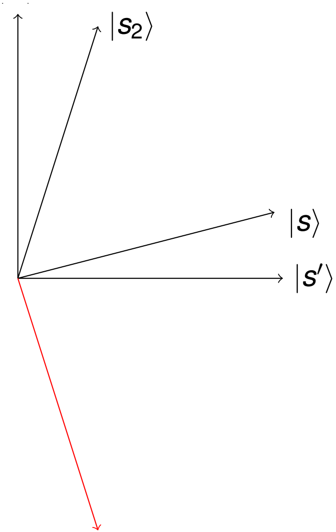

On applique $-H_s$, qui effectue la symmétrie par rapport à $|s\rangle$

# Dessin



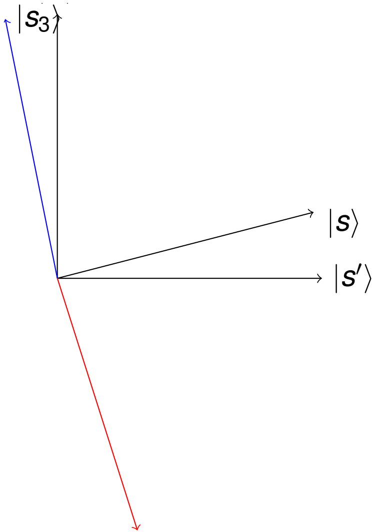

On obtient $|s_2\rangle$.

# Dessin



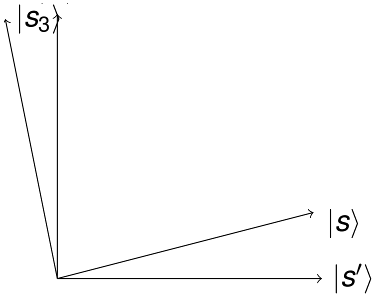

On continue.

# Dessin




On continue.

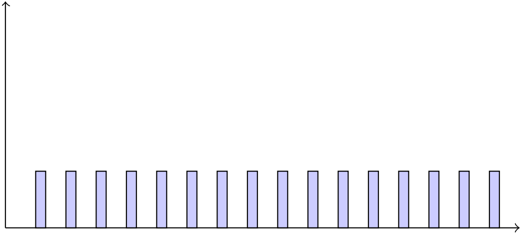
# Dessin




On continue.

# Amplitudes

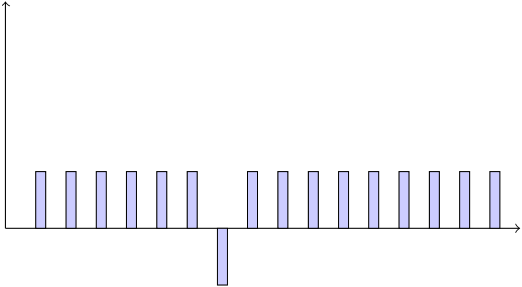
Dans l'espace des amplitudes (La colonne $i$ est l'amplitude de $|x_i\rangle$)



On part de $|s\rangle$, la superposition uniforme de tous les $|x\rangle$.

# Amplitudes

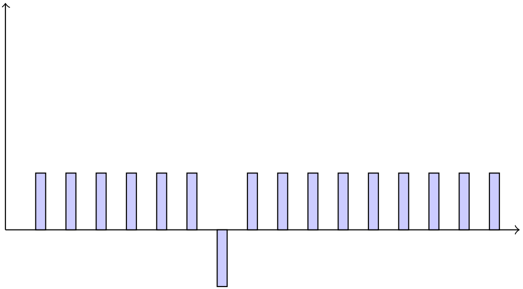
Dans l'espace des amplitudes (La colonne $i$ est l'amplitude de $|x_i\rangle$)



On applique $U_f$, qui inverse les $x$ tels que $f(x) = 1$.

# Amplitudes

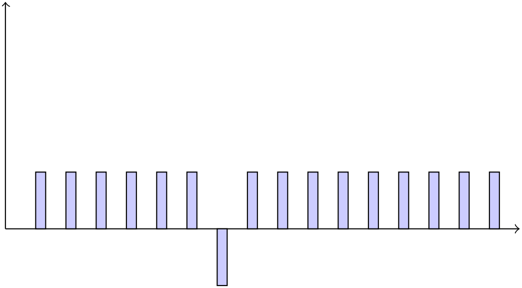
Dans l'espace des amplitudes (La colonne *i* est l'amplitude de $|x_i\rangle$)



On applique $-H_s$, qui effectue la symmétrie par rapport à $|s\rangle$

# Amplitudes

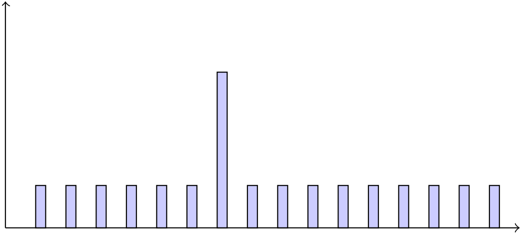
Dans l'espace des amplitudes (La colonne $i$ est l'amplitude de $|x_i\rangle$)



Cela revient à *inverser par rapport à la moyenne*

# Amplitudes

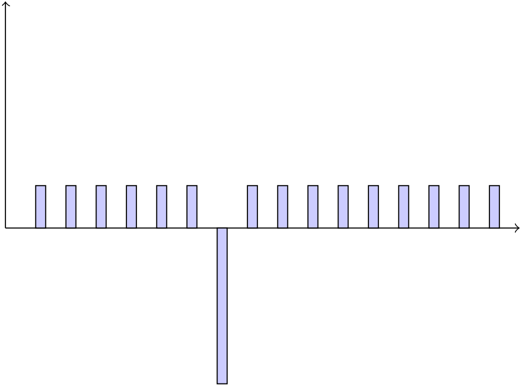
Dans l'espace des amplitudes (La colonne *i* est l'amplitude de $|x_i\rangle$)



On obtient $|s_1\rangle$. Si on mesure $|s_1\rangle$, on a augmenté la probabilité d'observer un $x$ tel que $f(x) = 1$.

# Amplitudes

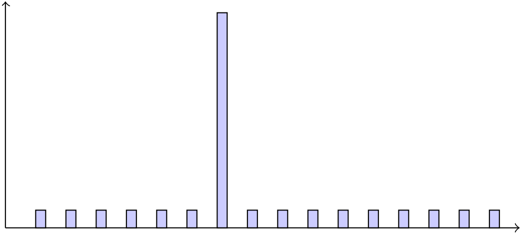
Dans l'espace des amplitudes (La colonne $i$ est l'amplitude de $|x_i\rangle$)



On applique $U_f$

# Amplitudes

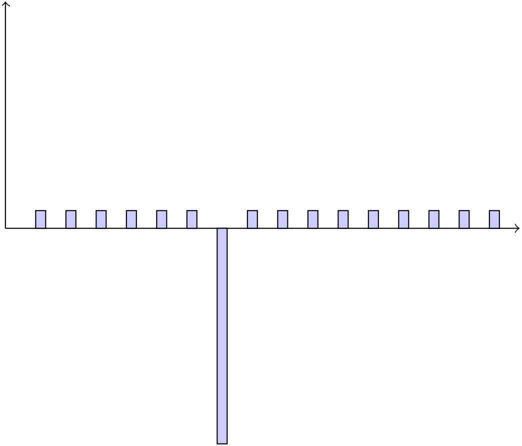
Dans l'espace des amplitudes (La colonne *i* est l'amplitude de $|x_i\rangle$)



On fait la symmétrie par rapport à la moyenne

# Amplitudes

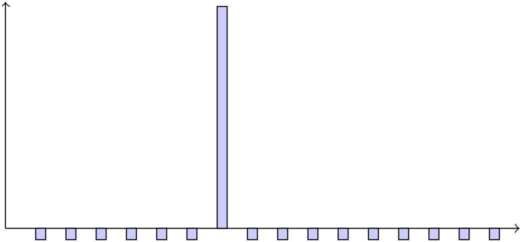
Dans l'espace des amplitudes (La colonne *i* est l'amplitude de $|x_i\rangle$)



On continue.

# Amplitudes

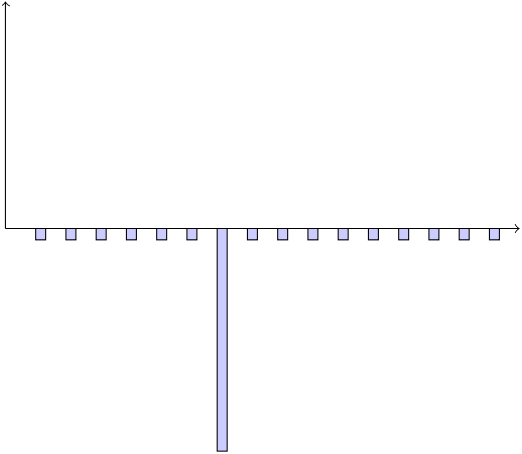
Dans l'espace des amplitudes (La colonne $i$ est l'amplitude de $|x_i\rangle$)



On continue.

# Amplitudes

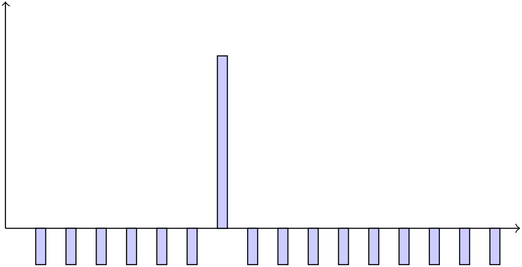
Dans l'espace des amplitudes (La colonne $i$ est l'amplitude de $|x_i\rangle$)



On continue.

# Amplitudes

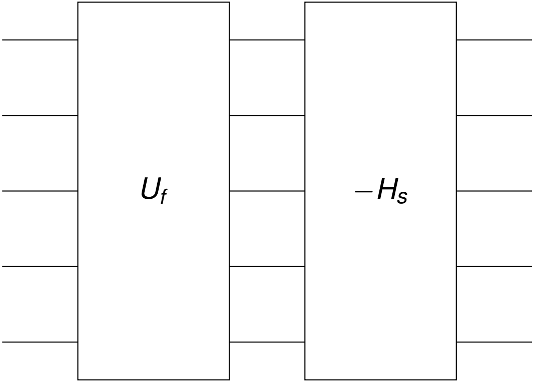
Dans l'espace des amplitudes (La colonne $i$ est l'amplitude de $|x_i\rangle$)



On continue.

# Principe de l'algorithme de Grover

Comment faire l'étape 2 ?



# Principe de l'algorithme de Grover

## Etape 1

Produire une superposition de toutes les entrées au problème.

## Etape 2

Modifier l'état du système de façon à augmenter les amplitudes des $x$ tels que $f(x) = 1$.

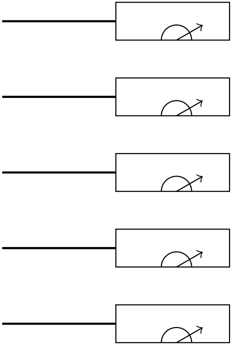
## Etape 3

Mesurer

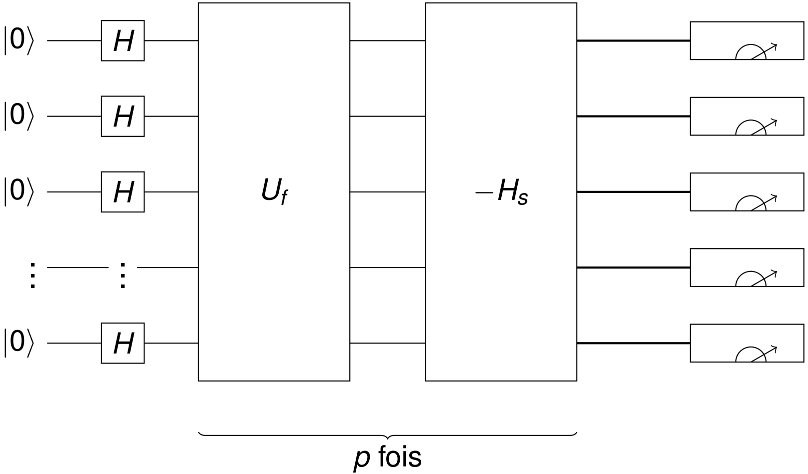
Comment faire l'étape 3 ?

# Principe de l'algorithme de Grover

Comment faire l'étape 3 ?



# Grover en entier



# Algorithme de Grover

Il reste deux choses à expliquer:

- Comment construire $H_s$
- Comment choisir $p$.

# Plan

# Construire $H_s$

Rappel:

- $|s\rangle$ est la superposition uniforme de tous les $|x\rangle$.
- $H_s$ est la reflexion de plan orthogonal à $|s\rangle$

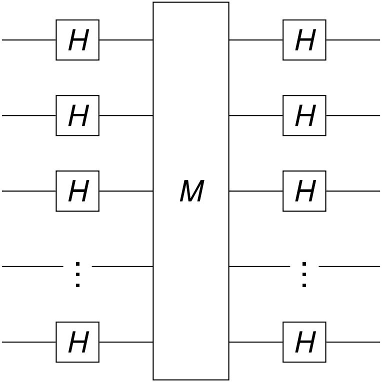
Dit autrement

- $H_s |s\rangle = - |s\rangle$
- $H_s |w\rangle = |w\rangle$ si $w$ est orthogonal à $s$.

# Construire $H_s$

Prenons un changement de base qui ramène $|s\rangle$ sur $|0\rangle$.
Dans ce cas, la matrice devient:

$$\begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$




Avec

$$M = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

# Construire $M$

Appliquons le changement $0 \leftrightarrow 1$ sur tous les bits, c'est à dire la matrice $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

La matrice devient:

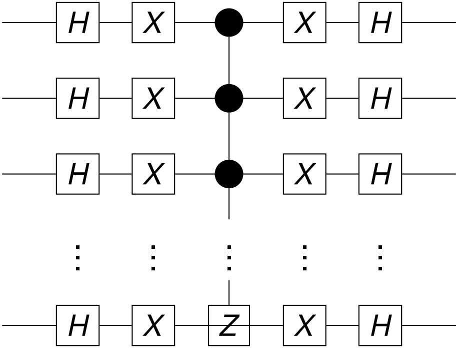
$$N = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Cette dernière matrice peut s'écrire ainsi:

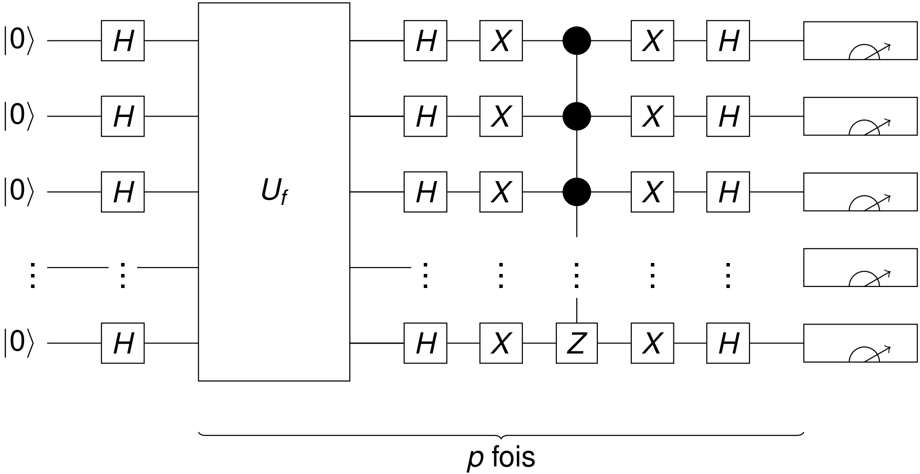
- $N \left| x_1 x_2 \dots x_{n-1} y \right\rangle = \left| y \right\rangle$ si l'un des $x_i \neq 1$
- $N \left| x_1 x_2 \dots x_{n-1} y \right\rangle = Z \left| y \right\rangle$ si tous les $x_i = 1$

Où $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.

La matrice $N$ est donc la matrice souvent appelée controle-controle-...-controle Z.


H
X
X
H
Z

# Grover en entier (bis)



# Plan

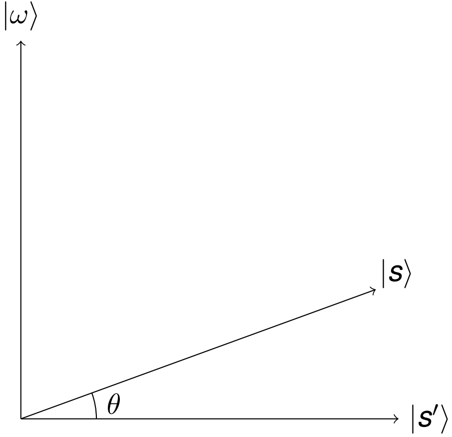
# Choisir les paramètres

Il reste à trouver combien de fois on répète l'opérateur dans Grover.

- Si on se trompe dans sa valeur, les $x$ tels que $f(x) = 1$ ne seront pas les plus probables!
- Dépend du nombre de solutions.
  - Soit on le connaît à l'avance, et on peut s'en servir
  - Soit il faut essayer de le deviner

# Dessin

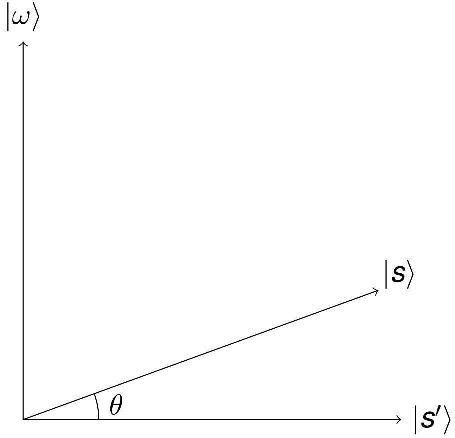
Dans l'espace $|\omega\rangle$ , $|s'\rangle$:




- On part de $|s\rangle$, on cherche à obtenir $|\omega\rangle$
- A chaque étape, on applique une réflexion d'axe $|s'\rangle$, puis d'axe $|s\rangle$, jusqu'à être proche de $|\omega\rangle$.
- On cherche combien de fois il faut appliquer les réflexions.

# Dessin

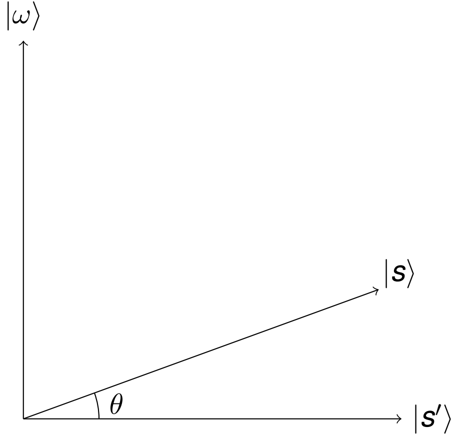
Dans l'espace $|\omega\rangle$ , $|s'\rangle$:




Soit $\theta$ l'angle entre $s$ et $s'$. Applique une réflexion d'axe $|s'\rangle$, puis d'axe $|s\rangle$ revient à appliquer une *rotation* d'angle $2\theta$.

# Dessin

Dans l'espace $|\omega\rangle, |s'\rangle$:




On cherche donc à résoudre l'équation $\theta + 2p\theta = \pi/2$.

# Récapitulatif

- Soit $\theta$ l'angle entre les vecteurs $|s'\rangle$ et $|s\rangle$
- On a
$$|s\rangle = \sqrt{\frac{K}{N}}\,|\omega\rangle + \sqrt{\frac{N-K}{N}}\,\left|s'\right\rangle$$
Et donc $\theta = \arcsin\sqrt{\frac{K}{N}}$
- Si on fait $p$ itérations de l'algorithme principal, on se retrouve en
$$\sin((1+2p)\theta)\,|\omega\rangle + \cos((1+2p)\theta)\,\left|s'\right\rangle$$
- La probabilité de lire un $x$ tel que $f(x) = 1$ est donc
$$\sin^2((1+2p)\theta)$$

# Plan

# Une solution

On cherche à maximiser

$$\sin^2\left((1+2p)\arcsin\sqrt{\frac{1}{N}}\right)$$

On veut donc

$$(1+2p)\arcsin\sqrt{\frac{1}{N}} = \pi/2$$

D'où

$$p = \frac{\pi}{4\arcsin\sqrt{\frac{1}{N}}} - 1/2$$

Pour $N$ grand:

$$p = \frac{\pi\sqrt{N}}{4} - 1/2$$

# Une seule solution

**Theorem**

*S'il y a une seule solution, il faut répeter l'étape 2 de Grover*

$$\frac{\pi}{4 \arcsin \sqrt{\frac{1}{N}}} - \frac{1}{2} = O\left(\sqrt{N}\right)$$

# Analyse de l'erreur

Soit $q$ le nombre d'étapes optimal, c'est à dire tel que

$$(1 + 2q)\theta = \pi/2$$

Et $p$ le nombre d'étapes réalisé en pratique (un entier)

$$p = q \pm 1/2$$

Donc

$$(1 + 2p)\theta = \pi/2 \pm \theta$$

La probabilité de trouver $x$ est donc au minimum:

$$\sin^2(\pi/2 - \theta) = 1 - \sin^2\theta = 1 - \frac{1}{N}$$

# Une seule solution

**Theorem**

*S'il y a une seule solution, il faut répeter l'étape 2 de Grover*

$$\frac{\pi}{4\arcsin\sqrt{\frac{1}{N}}} - \frac{1}{2} = O\left(\sqrt{N}\right)$$

*On obtient ainsi un* $x$ *tel que* $f(x) = 1$ *avec probabilité au moins* $1 - \frac{1}{N}$

# $K$ solutions

**Theorem**

*S'il y a exactement $K$ solutions, avec $K << N$, il faut répeter l'étape 2 de Grover*

$$\frac{\pi}{4 \arcsin \sqrt{\frac{K}{N}}} - \frac{1}{2} = O\left(\sqrt{\frac{N}{K}}\right)$$

*On obtient ainsi un $x$ tel que $f(x) = 1$ avec probabilité au moins $1 - \frac{K}{N}$.*

# Cas particulier

Dans le cas particulier $K = N/4$, on trouve

$$p = \frac{\pi}{4 \arcsin \sqrt{\frac{1}{4}}} - 1/2 = \frac{pi}{4\pi/6} - 1/2 = 1$$

Il suffit donc de prendre $p = 1$, d'appeler une fois Grover, et on aura le résultat *sans erreur*

# Plan

# Nombre de solutions inconnues

Comment faire si le nombre de solutions est inconnu ?

- On peut se tromper un peu sur le nombre de solutions:
  - Si $K$ est le vrai nombre de solutions, et qu'on l'estime à $L = K \pm \epsilon$ la probabilité d'erreur change très peu
  - En pratique, la connaître à un facteur 2 peut être suffisant
- Mais il faut quand même avoir une petite idée, ou savoir s'en sortir sans la connaître!

# Méthode 1 - Borne sur le nombre de solutions

Si on suppose qu'il y a au moins $L$ solutions mais pas plus de $N/2$, il suffit de tirer $p$ aléatoirement entre 0 et $\sqrt{\frac{N}{L}}$.

(Vrai également si le nombre de solutions est inférieur à $3N/4$ en faisant un peu plus attention aux détails.)

# Preuve

Soit $K$ le vrai nombre de solutions et $\theta = \arcsin\sqrt{\frac{K}{N}}$. Posons $M = \sqrt{\frac{N}{L}}$.
Si on fait $p$ exécutions de la boucle, la probabilité de succès est $\sin^2((2p+1)\theta)$
Donc en moyenne, si on tire $p$ aléatoirement entre 0 et $M-1$ la proba de succès est

$$\frac{1}{M}\sum_{p=0}^{M-1}\sin^2((2p+1)\theta) = \frac{1}{M}\sum_{p=0}^{M-1}\frac{(1-\cos(2p+1)2\theta)}{2} = \frac{1}{2} - \frac{sin(4M\theta)}{4M\sin(2\theta)}$$

# Preuve

La proba de succès est

$$\frac{1}{2} - \frac{sin(4M\theta)}{4Msin(2\theta)}$$

Si $\theta < \pi/4$ alors $\sin(2\theta) \geq \sin(\theta) \geq \sqrt{\frac{K}{N}}$
Donc

$$Msin(2\theta) \geq M\sqrt{\frac{K}{N}} \geq \sqrt{\frac{N}{L}}\sqrt{\frac{K}{N}} \geq \sqrt{\frac{K}{L}} \geq 1$$

Dans ce cas

$$\frac{1}{2} - \frac{sin(4M\theta)}{4M\sin(2\theta)} \geq \frac{1}{2} - \frac{1}{4} \geq \frac{1}{4}$$

# Méthode 1 - Borne sur le nombre de solutions

Si on suppose qu'il y a au moins $L$ solutions mais pas plus de $N/2$, il suffit de tirer $p$ aléatoirement entre 0 et $\sqrt{\frac{N}{L}}$.

L'algorithme a une proba de réussite de $1/4$, on peut ensuite l'itérer pour augmenter cette probabilité.

# Méthode 2

- Défaut de la méthode précédente: si $L$ est très différent de $K$, on a un algo de complexité $\sqrt{\frac{N}{L}}$ au lieu de $\sqrt{\frac{N}{K}}$
- Au lieu de supposer avoir une borne sur le nombre de solutions, on va essayer de deviner petit à petit le nombre de solutions.

# Méthode 2

- Commencer à $M = 1$
- Tirer $p$ aléatoirement entre 0 et $M - 1$ et exécuter Grover qui nous donne un $x$
- Si $f(x) = 1$ c'est gagné, sinon $M = 1.65M$.

Le temps moyen avant de trouver la solution est $O(\sqrt{\frac{N}{K}})$. Preuve dans Boyer-Brasser-Hoyer-Tapp 1996.

# Plan

# Applications directes

## Problèmes NP

- Soit $\phi$ une formule 3CNF, trouver $S$ tel que $\phi[S] = 1$.
  - Grover en $\sqrt{2^n} = 2^{n/2} = 1.414^n$.
  - Meilleur algo classique en $1.308^n$.
- Trouver une 3-coloration d'un graphe
  - Grover en $\sqrt{3^n} = 1.732^n$ naivement
  - Meilleur algo classique en $1.329^n$
- Etant donné $x, y$, trouver $K$ tel que $DES(K, x) = y$
  - Applications en crypto

# Plan

# Collisions

Soit $f : \{0,1\}^n \to \{0,1\}^m$. On suppose que chaque élément a exactement $k \geq 2$ préimages.
On cherche $x \neq y$ tel que $f(x) = f(y)$.

Application typique: Cryptographie
Comment utiliser Grover ?

# Collisions - Algo 1

On fixe $x_0$ et on cherche $x \neq x_0$ tel que $f(x) = f(x_0)$

- L'algorithme est donc en complexité $O\left(\sqrt{\frac{N}{k-1}}\right)$.

# Collisions - Algo 2

Soit $g(x, y) = 1$ si $f(x) = f(y)$ et $x \neq y$.
On cherche $x, y$ tels que $g(x, y) = 1$.

- Il y a $N^2$ couples $(x, y)$ différents
- Pour chaque $x$, on a $k - 1$ différents $y$ qui sont solutions, donc $N(k - 1)$ solutions en tout
- L'algorithme est donc en complexité $O\left(\sqrt{\frac{N^2}{N(k-1)}}\right) = O\left(\sqrt{\frac{N}{k-1}}\right)$.

# Collisions

Au fait, quelle est la complexité de l'algo classique ? (en nombre de requêtes)

# Paradoxe des anniversaires

- Choisir $2\sqrt{N}$ valeurs de $x$ parmi les $N$ possibles
- Avec grande probabilité, parmi ces $2\sqrt{N}$ valeurs, il y a une collision
- On a donc fait seulement $2\sqrt{N}$ requêtes.

Le temps total est en $O(\sqrt{N}\log N)$ avec un algo de tri par exemple.

# Collisions - Le bon algo

## Brassard-Hoyer-Tapp

Soit $M = \sqrt[3]{N}$

- Choisir (n'importe comment) un ensemble $S$ de taille $M$ et vérifiez qu'il n'y a pas de collision dedans.
    - Complexité: $M$ requêtes à $f/U_f$.
- S'il y a une collision, c'est gagné. Sinon, soit $T$ l'ensemble des $f(x)$ obtenus
- Construire la fonction $g$, définie sur $\{0,1\}^n \setminus S$, telle que $g(y) = 1$ si $f(y) \in T$.
- Utiliser Grover pour trouver un $y$ tel que $g(y) = 1$.
    - On cherche une solution dans un espace de taille $N - M \simeq N$
    - Le nombre de solutions est au moins $M$
    - Complexité $\sqrt{\frac{N}{M}} = \sqrt[3]{N}$ requêtes à $f$.

Nombre total de requêtes: $2\sqrt[3]{N}$.

# Plan

# Minimum

Soit $f : \{0, 1\}^n \to \mathbb{N}$. On cherche le minimum de $f$

Comment utiliser Grover ?

Quel oracle doit-on prendre ?

# Minimum

Soit $g(x, y) = 1$ si $f(x) < f(y)$, 0 sinon. On utilise l'oracle $U_g$ pour trouver le minimum.

Algo classique en O(n) (évident).

# Quickselect

Un algo probabiliste pour trouver le minimum dans un tableau $T$ de taille $n$:

- Si $n = 1$, renvoyer $T[0]$
- Sinon tirer aléatoirement une case $i$ du tableau
- Soit $T'$ les éléments de $T$ inférieurs à $T[i]$. Appeler récursivement l'algo sur $T'$.

Complexité ?

Note: se généralise (et devient utile) pour trouver le $k$-ème plus petit élément du tableau.

Intuition:

- Si on tape au milieu du tableau à chaque étape, la taille du tableau diminue de moitié à chaque étape
- Complexité

$$n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + ... = 2n$$

La complexité de l'algo est en $2n$.

Réalité. Soit $X_n$ le temps moyen sur un tableau de taille $n$

- Avec proba 1/2, la taille du tableau diminue d'au moins la moitié.
- D'où

$$X_n \leq n + \frac{1}{2}X_{n/2} + \frac{1}{2}X_n$$

D'où $X_n \leq 4n$. La complexité de l'algo est donc inférieure à $4n$.

En supposant Grover infaillible:

- Soit $x \in \{0, 1\}^n$ tiré au hasard
- Trouver avec Grover s'il existe $y$ tel que $f(y) < f(x)$. Si non, on a gagné
- Si oui, poser $x = y$ et recommencer

Complexité ?

# Analyse

- Soit $N = \{0,1\}^n$
- Soit $X_k$ le temps moyen s'il y a $k$ éléments plus petits que l'élément courant (Au départ $k = N$).
- A $k$ donné, Grover prend un temps $\sqrt{\frac{N}{k}}$.
- Avec proba $1/2$, $k$ est divisé par 2.
- D'où

$$X_k \leq \sqrt{\frac{N}{k}} + \frac{1}{2}X_k + \frac{1}{2}X_{k/2}$$

D'où

$$X_N \leq 2\sqrt{1} + 2\sqrt{2} + 2\sqrt{4} + \dots 2\sqrt{N}$$

$$X_N \leq 7\sqrt{N}$$

# En pratique

L'algo de Grover n'est pas infaillible:

- il ne nous dira jamais "il n'y a pas de solution".
- il a une petite probabilité d'erreur

On doit fixer à l'avance le critère d'arrêt

- Dans l'algo infaillible, on effectue $7\sqrt{N}$ appels à l'oracle.
- Si on s'arrête après $14\sqrt{N}$ appels, d'après l'inégalité de Markov, on a une proba de succès d'au moins $1/2$.
- Si on tient compte des erreurs dans Grover, il faut plutôt prendre $28\sqrt{N}$.

# Plan

# Mise en abyme

On peut utiliser l'algorithme de Grover pour accélérer n'importe quel algo de recherche

# Quantum Amplitude Amplification

## Brassard-Hoyer-Mosca-Tapp

### Theorem

*Soit $\mathcal{A}$ un algo quantique sans mesure qui fait $K$ requêtes et qui trouve (après mesure) avec proba $p$ un élément $x$ qui vérifie une certaine propriété.*
*Alors on peut le transformer en un algo quantique qui, après mesure, a une proba $1/2$ de trouver $x$ en $O(K\sqrt{1}p)$ requêtes.*

# Exemple

$f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$
On cherche $x$ tel que pour tout $y$, $f(x,y) = 1$

On note $N = \{0,1\}^n$, donc l'algo naif est en $O(N^2)$.

# Méthode 1

recherche-Grover de $x$ tel que pour tout $y$, $f(x, y) = 1$.

- Soit $g$ la fonction tel que $g(x) = 1$ si pour tout $y$, $f(x, y) = 1$.
- $g(x)$ se calcule en $O(N)$ appels à l'oracle.
- Donc trouver $x$ tel que $g(x) = 1$ se fait en $O(N\sqrt{N})$ appels.

# Méthode 2

Pour tout $x$, on fait Grover pour savoir si pour tout $y$, $f(x, y) = 1$.

- A $x$ donné, soit $g_x$ la fonction tel que $g_x(y) = 1 - f(x, y)$
- $g_x(y)$ se calcule en $O(1)$ appels à l'oracle.
- Pour tout $x$, on cherche un $y$ tel que $g_x(y) = 1$. Si on n'en trouve pas, alors $x$ est solution du problème
- $O(N\sqrt{N})$ appels

# Méthode 3

Grover dans Grover:

- A $x$ donné, considérons le problème de trouver s'il existe $y$ tel que $f(x, y) = 0$.
- En utilisant Grover, on a un algorithme quantique pour ce problème en $\sqrt{N}$ requêtes et proba de succès quasi égale à 1.

- Considérons maintenant le problème de trouver un $x$ tel que $\forall y, f(x, y) = 1$.
- On a un algo quantique avec $\sqrt{N}$ requêtes et proba de succès $1/N$ pour ce problème : choisir $x$ aléatoirement puis appliquer Grover.
- Donc en amplifiant cet algorithme, on a un algo avec $O(\sqrt{N} \times \sqrt{N}) = O(N)$ requêtes.

(en pratique $O(N \log N)$)

# Plan

# Algorithme de Grover

Soit $f : \{0,1\}^n \to \{0,1\}$ une fonction. On note $N = 2^n$.

L'algorithme de Grover trouve $x$ tel que $f(x) = 1$ en temps $O(\sqrt{N})$ et avec $O(\sqrt{N})$ appels à $U_f$.

Rappel: $U_f \left|x\right\rangle = (-1)^{f(x)} \left|x\right\rangle$

- Beaucoup d'applications potentielles
- Faire attention au temps pour calculer $U_f$.