A - III - (4) Quantum searching - Grover's algo.

Previously we saw that we can determine the period of a function exponentially faster with quantum queries than with classical queries (if the function itself is efficiently computable, for ex: modular expo. fct → Shor's algo).

In this section we will consider a pb for which the quantum speed up is not exponential but only polynomial (quadratic to be precise).

→ this is the search pb

and the reason this is an interesting pb is because of the large range of applications of this pb.

ex: Suppose you are given a map containing many cities and want to determine the shortest route passing through all the cities -
Possible algo: search all possible route and keep a record of which one has the shortest length. On a classical computer: if there are N possible routes → O(N) operations -

But there is a quantum search algo: "Grover's algorithm" which requires only $O(\sqrt{N})$ operations - And this algo can be applied to many search pbs beyond route finding.

* The quantum search pb:
Suppose we are searching for one element among a space of $N = 2^n$ elements. (integers or n-bit strings).

Let us denote by w the element we are looking for - ↳ "the marked string". "winner"

We want to formalize this search pb in the black box setting :

Suppose we have a function f_w that can recognize the solution w :

$$f_w : \{0,1\}^n \longrightarrow \{0,1\}$$
$$x \longmapsto f(x)$$

with $f_w(x) = \begin{cases} 1 \text{ if } x = w \\ 0 \text{ if } x \neq w \end{cases}$

$\Rightarrow f_w$ is a verifier.

This can be formulated as quantum black box pb where the oracle computes the unitary

$$U_w : |x\rangle \otimes |y\rangle \longmapsto |x\rangle \otimes |y \oplus f_w(x)\rangle$$

($|x\rangle$: n-qubit, $|y\rangle$: 1-qubit)

Classically → we need to query more than $\frac{N}{2}$ times to find w with proba $> \frac{1}{2}$.

Note: this is the black box version of an NP-hard pb, where there is a unique witness accepted by the circuit, but the pb has no structure, so there is no other choice than to do exhaustive searching.

Quantum search → apply the oracle with the extra qubit in superposition state $|-\rangle$ (as in Deutsch-Josza algo) to turn U_w into a phase oracle (usual "trick").

$$U_w : |x\rangle \otimes |-\rangle \longmapsto |x\rangle \otimes \frac{1}{\sqrt{2}} \left(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle \right)$$

$$= |x\rangle \otimes \begin{cases} |-\rangle \text{ if } f(x) = 0 \\ -|-\rangle \text{ if } f(x) = 1 \end{cases}$$

$$= (-)^{f(x)} |x\rangle \otimes |-\rangle$$

$$= \begin{cases} - |x\rangle \otimes |-\rangle & \text{if } x = w \\ + |x\rangle \otimes |-\rangle & \text{if } x \neq w \end{cases}$$

U_ω effectively acts on $|x\rangle$ as

$$U_\omega = \mathbb{1} - 2|\omega\rangle\langle\omega|$$

(check: $U_\omega |x\rangle = \mathbb{1}|x\rangle - 2|\omega\rangle\underbrace{\langle\omega|x\rangle}$

$= |x\rangle - 2|\omega\rangle \delta_{\omega,x}$

$= \begin{cases} |x\rangle \text{ if } x \neq \omega \\ |\omega\rangle - 2|\omega\rangle = -|\omega\rangle \\ \quad \text{if } x = \omega \end{cases}$)

Now we can express any arbitrary n-qubit state $|\psi\rangle$ as a superposition of $|\omega\rangle$ + state orthogonal to $|\omega\rangle$:

$$|\psi\rangle = \overset{\sin\theta}{a}|\omega\rangle + \overset{\cos\theta}{b}|\omega^\perp\rangle$$

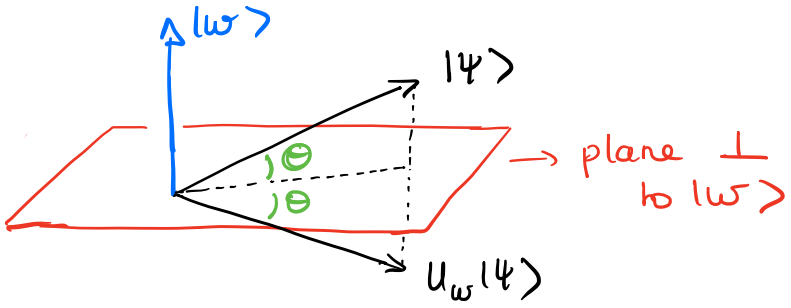
↳ with: $\langle\omega|\omega^\perp\rangle = 0$

Then U_w acts on $|\psi\rangle$ as:

$$U_w : |\psi\rangle \longmapsto \underbrace{a}_{\sin\theta}\underbrace{U_w|w\rangle}_{-|w\rangle} + \underbrace{b}_{\cos\theta}\underbrace{U_w|w^\perp\rangle}_{+|w^\perp\rangle}$$

$$= \underbrace{-a}_{\sin\theta}|w\rangle + \underbrace{b}_{\cos\theta}|w^\perp\rangle .$$

One can think about this transformation U_w geometrically as a reflection of the vector $|\psi\rangle$ about the hyperplane orthogonal to $|w\rangle$.



* Grover's algorithm :

(1) State preparation :

In practice we will start with $|\Psi\rangle$ in uniform superposition of all computational-basis states

$$|\Psi\rangle = |s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

as it is easy to prepare with a tensor product of Hadamard gates $H^{\otimes n}$: $|s\rangle = H^{\otimes n} |0\rangle^{\otimes n}$.

In our previous notations :

$$|s\rangle = \underbrace{\frac{1}{\sqrt{N}} \sum_{x \neq w} |x\rangle}_{\cos\theta\, |w_1\rangle} + \underbrace{\frac{1}{\sqrt{N}} |w\rangle}_{\sin\theta\, |w\rangle}$$

with $|w_1\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle \Rightarrow \cos\theta = \sqrt{\frac{N-1}{N}}$

and $\sin\theta = 1/\sqrt{N}$.

$\Rightarrow |s\rangle$ contains the component $|w\rangle$ that we are looking for:

overlap $\langle w|s\rangle = \frac{1}{\sqrt{N}} = \sin\theta$

$\boxed{\theta = \arcsin\left(1/\sqrt{N}\right) \approx 1/\sqrt{N}}$ (N large)

Note: if we were to measure now

$\Rightarrow$ the n-qubit state would collapse to one of the N computational-basis states with proba $\left(1/\sqrt{N}\right)^2 = 1/N$ $\Rightarrow$ chances of obtaining the winner w is one in $N = 2^n$.

$\Rightarrow$ on average would need to try $\sim \frac{N}{2} = 2^{n-1}$ times to get w. (as classically).

(2) Amplitude Amplification :

which is how the quantum computer will enhance significantly this proba while suppressing the proba of obtaining $|x\rangle \neq |w\rangle$.

To do that we apply many times successively the "Grover iteration" :

$$U_{Grover} = U_s U_w$$

where
$$\begin{cases} U_w = \mathbb{1} - 2|w\rangle\langle w| \\ U_s = 2|s\rangle\langle s| - \mathbb{1} \end{cases}$$

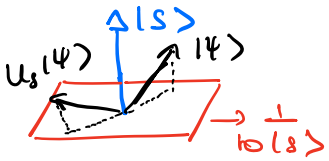
U_w : the query, (as discussed above)

↳ reflects a state vector about the axis determined by $|s\rangle$.

$U_s|s\rangle = |s\rangle$ → leaves $|s\rangle$ unchanged

$U_s|s^\perp\rangle = -|s^\perp\rangle$ → flips phase

↳ perp to $|s\rangle$



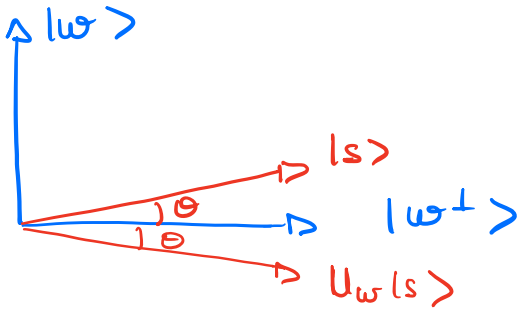
What does $U_{Grover} = U_s U_w$ do ?

There is a nice geometrical interpretation as the two reflections U_s and U_w generate a rotation in a two-dim plane.

Since U_s reflects about $|s\rangle$ and U_w reflects about plane $\perp |w\rangle$ $\rightarrow$ if we start with state $|s\rangle$ $\rightarrow$ we will stay in the 2-d plane spanned by $|w\rangle$ and $|s\rangle$.

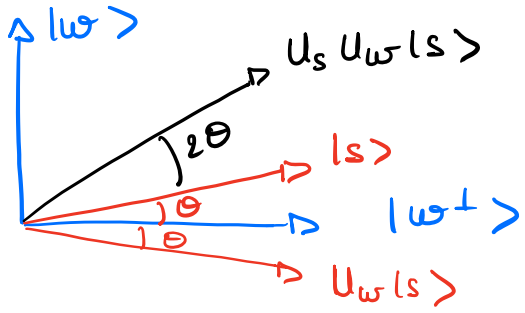
$\Rightarrow$ Action of U_{Grover} in this plane :

- Start with $|s\rangle = (1/\sqrt{N}) \sum_x |x\rangle$ $\Rightarrow$ amplitude of all $|x\rangle$ is $1/\sqrt{N} \sim \theta$.
- Apply U_w = reflection about hyperplane $\perp$ to $|w\rangle$



Now amplitude of $|w\rangle = -1/\sqrt{N}$ ($= -\sin\theta$), $|x \neq w\rangle = 1/\sqrt{N}$ ($= \cos\theta$)

• Apply U_s = reflection about $|s\rangle$:



⇒ we have rotated $|s\rangle$ closer to $|w\rangle$ by angle 2θ.

The new state $U_s U_w |s\rangle$ has a larger overlap with $|w\rangle$, smaller overlap with $|w^\perp\rangle$.

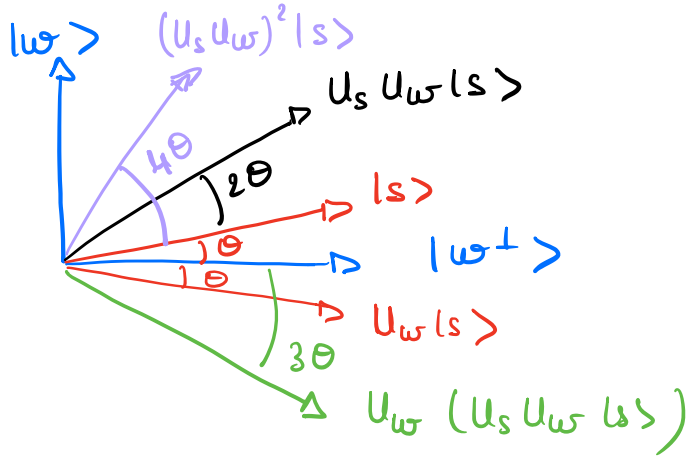
Amplitude of $|w\rangle$ $= \langle w | U_s U_w | s\rangle$
$= \sin 3\theta > \sin\theta$

Amplitude of $|w^\perp\rangle$
$= \cos 3\theta < \cos\theta$

$\sin\theta \sim \theta - \frac{\theta^3}{3!} + \cdots$

$\cos\theta \sim 1 - \frac{\theta^2}{2!} \cdots$

Then we repeat this procedure and apply $U_s U_w$ again to get us even closer to $|w\rangle$.



$\Rightarrow$ etc---

Each Grover iteration rotates the state vector by angle 2θ towards $|w\rangle$.

$\Rightarrow$ After t iterations we will be in state

$$|\psi_t\rangle = (U_s U_w)^t |s\rangle$$
$$= \sin(\theta + 2t\theta)\ |w\rangle + \cos(\theta + 2t\theta)\ |w^\perp\rangle$$

(3) Measurement (in computational basis).

$\Rightarrow$ will find state $|w\rangle$ with proba

$$\text{proba}(w) = \sin^2[(2t+1)\theta]$$

→ How many times do we need to apply the Grover iteration, to find outcome $|w\rangle$ with proba $O(1)$?

⇒ we want to determine t such that

$$(2t+1)\,\theta \approx \pi/2$$

in that case our state vector would be approximately aligned with $|w\rangle$ and the outcome of the measurement would be $|w\rangle$ with probability ≈ 1.

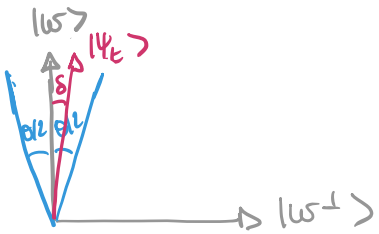
$$\Rightarrow t \approx \left(\frac{\pi}{2} - \theta\right)\frac{1}{2\theta} \approx \frac{\pi}{4\theta} - \frac{1}{2}$$

and $\theta \approx 1/\sqrt{N}$ $\Rightarrow$ $\boxed{t \approx T = \frac{\pi}{4}\sqrt{N}}$ (neglect $-1/2$ because $\sqrt{N}$ large)

It works because probabilities are squares of amplitudes in quantum mechanics –
⇒ if amplitude ↑ linearly, the proba ↑ quadratically.

classically: success proba ↑ linearly with the # of queries.

Note : To be precise : if we choose t such that $(2t+1)\theta = \frac{\pi}{2} + \delta$ where $|\delta| \leq \frac{\theta}{2} \approx \frac{1}{2\sqrt{N}}$



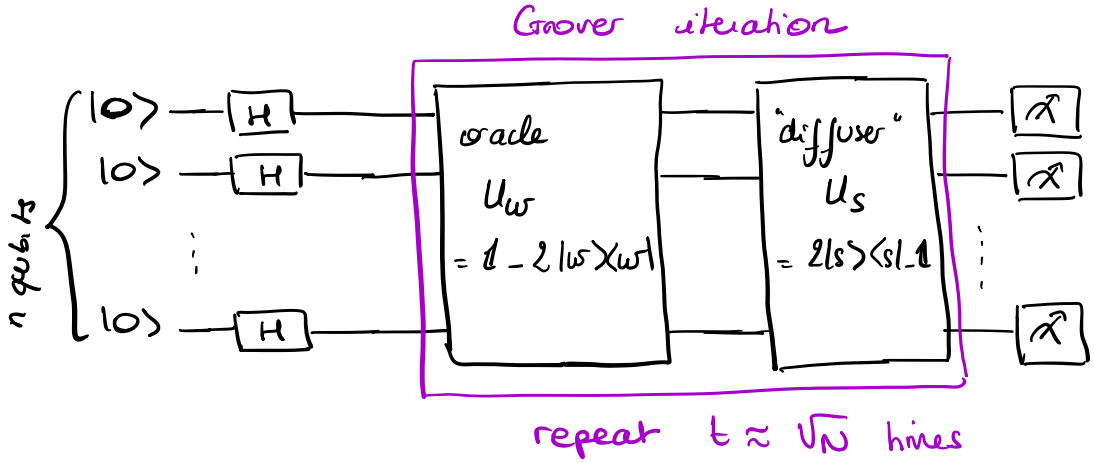
$$\Rightarrow |\psi_t\rangle = (U_s U_\omega)^t |s\rangle$$
$$= \cos\delta\,|\omega\rangle + \sin\delta\,|\omega^\perp\rangle$$

when we measure in the computational basis we find outcome $|\omega\rangle$ with proba

$$\text{proba}(\omega) = (\cos\delta)^2 \geq (\cos\theta/2)^2 \approx 1 - \left(\frac{\theta}{2}\right)^2 \approx 1 - \frac{1}{4N}$$

which is $O(1)$.

$\Rightarrow$ Quantum circuit : (only show the n-qubit reg)



Quantum searching – Grover's algo. (continued)

A few remarks :

* Oracle U_{f_0}: the description of the oracle without describing how it works in practice is a bit abstract and can be puzzling -

It seems that the oracle already knows the answer to the search pb => what would be the pb ?

However it is not the case – There is a distinction between knowing the solution to a search pb and being able to recognize the solution – the crucial pb is that it is possible to do the latter without being able to do the former – (as discussed before, one can verify a solution efficiently without knowing how to obtain the sol° efficiently (ex factoring)).

The oracle is really a verifier.

* Diffuser U_S :

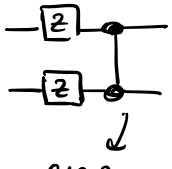
U_S should be applicable efficiently -
This is indeed the case as

$$U_S = 2|s\rangle\langle s| - \mathbb{1}$$
$$= 2\, H^{\otimes n} |0\rangle^{\otimes n} \langle 0| H^{\otimes n} - H^{\otimes n} H^{\otimes n}$$
$$= H^{\otimes n} \underbrace{\left[2|0\rangle^{\otimes n}\langle 0| - \mathbb{1} \right]}_{U_0} H^{\otimes n}$$

$\Rightarrow U_0$ leaves $|0\rangle^{\otimes n}$ unchanged
but flips the phase of any other state

This can be implemented with $C^{(n-1)}(Z)$ and 1-qubit gates.

Ex: 2 qubits $\quad U_0 = C(Z)\,(Z \otimes Z) =$



$C(Z)$
(symmetric)

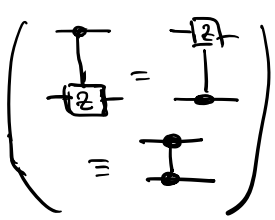
$U_0 : |00\rangle \mapsto |00\rangle \mapsto |00\rangle$

$U_0 : |01\rangle \mapsto -|01\rangle \mapsto -|01\rangle$

$U_0 : |10\rangle \mapsto -|10\rangle \mapsto -|10\rangle$

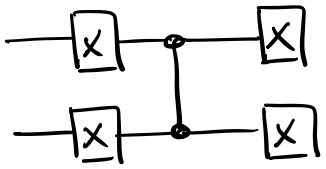
$U_0 : |11\rangle \mapsto +|11\rangle \mapsto -|11\rangle$

$\rightarrow$ OK



or $U_0 = \ominus (X \otimes X)\ C(Z)\ (X \otimes X)$

$=$ $\ominus$



$|00\rangle \mapsto^{\ominus} |11\rangle \mapsto^{\ominus} |-11\rangle \mapsto^{+} -|00\rangle$

$|01\rangle \mapsto^{\ominus} |10\rangle \mapsto^{\ominus} |10\rangle \mapsto^{\ominus} |01\rangle$

$|10\rangle \mapsto^{\ominus} |01\rangle \mapsto^{\ominus} |01\rangle \mapsto^{\ominus} |10\rangle$

$|11\rangle \mapsto^{\ominus} |00\rangle \mapsto^{\ominus} |00\rangle \mapsto^{\ominus} |11\rangle$

global phase can be dropped.

More generally, a multi-controlled Z gate $C^{(n-1)}(Z)$ inverts the phase of the state $|11\ldots1\rangle$ while keeping other computational-basis states unchanged.

$$C^{(n-1)}(Z) = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & -1 \end{pmatrix}$$

Instead we want to apply a phase to $|11\ldots1\rangle$ and all computational basis states $\neq |00\ldots0\rangle$ while keeping $|0\ldots0\rangle$ unchanged.

This can be done by:

$$U_0 = (-)\, X^{\otimes n}\; C^{(n-1)}(Z)\; X^{\otimes n}$$

↳ can be dropped (gives global phase)

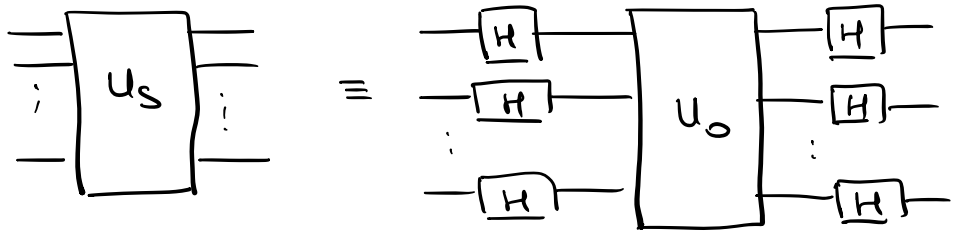
$C^{(n-1)}(Z)$ can be constructed from $C^{(n-1)}(X)$ where the target has H gate before and after and we know that $C^{(n-1)}(X)$ can be constructed from $O(n)$ Toffoli gates.

And a Toffoli itself can be constructed from a few 1- and 2- qubit gates

$\Rightarrow$ Overall:

U_S can be realized with circuit size $O(n)$.

$\Rightarrow$ diffuser U_S =



- So far we have considered that there is only one "winner" / "marked string" w -

What if there are $m \geq 2$ winners ?

Finding one winner is still a hard pb if $m \ll N$ as classically we need $O(N/m)$ queries to find any winner with proba $O(1)$.

Quantumly : $|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ can be

written in term of the (normalized) state

$$|W\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} |w_i\rangle$$

↳ superposition of all possible solutions (winners) $|w_i\rangle$

as : $|s\rangle = \cos\theta\, |W^\perp\rangle + \sin\theta\, |W\rangle$

where $|W^\perp\rangle = \frac{1}{\sqrt{N-m}} \sum_{x \neq \text{all } w_i} |x\rangle$

$$\Rightarrow \quad \sin\theta = \langle w | s \rangle$$

$$= \frac{1}{\sqrt{MN}} \sum_{i=1}^{m} \sum_{x=0}^{N-1} \underbrace{\langle w_i | x \rangle}_{\delta_{w_i, x}}$$

$$= \frac{1}{\sqrt{MN}} \underbrace{\sum_{i=1}^{m} 1}_{m}$$

$$= \sqrt{m/N}$$

and $\cos\theta = \langle w^{\perp} | s \rangle$

$$= \frac{1}{\sqrt{N-M}} \frac{1}{\sqrt{N}} \sum_{x} \underbrace{\sum_{x' \neq w_i} \langle x' | x \rangle}_{\underbrace{\sum_{x' \neq w_i} 1}_{N-M}}$$

$$= \sqrt{\frac{N-M}{N}}$$

$$= \sqrt{1 - \sin^2\theta}$$

Then the procedure is basically the same as in the case $m = 1$, i.e. we apply $U_G = U_s U_w$ several times.

U_w reflects about the hyperplane $\perp$ to $|w\rangle$

⇒ A state near $|w\rangle$ is obtained after T iterations such as

$$(2T+1)\,\theta = \pi/2$$

and the measurement will sample uniformly from the winners $|w_i\rangle$ (will pick one $|w_i\rangle$ among the m of them randomly).

The difference is that $\sin\theta = \sqrt{\frac{m}{N}}$

$$\Rightarrow \theta = \arcsin\left(\sqrt{\frac{m}{N}}\right) \approx \sqrt{\frac{m}{N}} \quad (\text{if } m \ll N)$$

$$\Rightarrow T = \frac{\pi}{4\theta} - \frac{1}{2} = \frac{\pi}{4\arcsin\left(\sqrt{\frac{m}{N}}\right)} - \frac{1}{2}$$

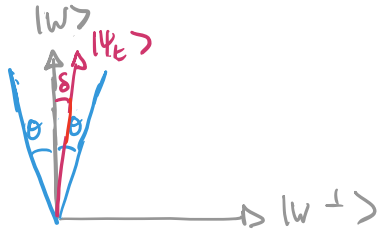
$$\boxed{\Rightarrow T = \frac{\pi}{4}\sqrt{\frac{N}{m}}} \quad (m \ll N)$$

⇒ even if there are more than one solutions the number of quantum queries needed to find a solution is

$$O\left(\sqrt{\#\text{ classical queries}}\right)$$

Again will have a small proba of error here due to the fact that T not exactly integer $\rightarrow$ not going to end up exactly on $|W\rangle$:

The optimal # of iterations t = closest integer to T will bring us to a state $|\psi_t\rangle$ which is within an angle δ of $|W\rangle$ with $|\delta| \leq \theta$.



$$\Rightarrow |\psi_t\rangle = (U_s U_w)^t |s\rangle$$
$$= \cos\delta\,|W\rangle + \sin\delta\,|W^\perp\rangle$$

when we measure in the computational basis we find outcome one solution $|w_i\rangle$ with proba :

$$\text{proba}(W) = (\cos\delta)^2 \geq (\cos\theta)^2$$
$$\approx \left(1 - \frac{\theta^2}{2}\right)^2$$
$$\approx 1 - \theta^2 \approx 1 - \left(\frac{m}{N}\right)$$

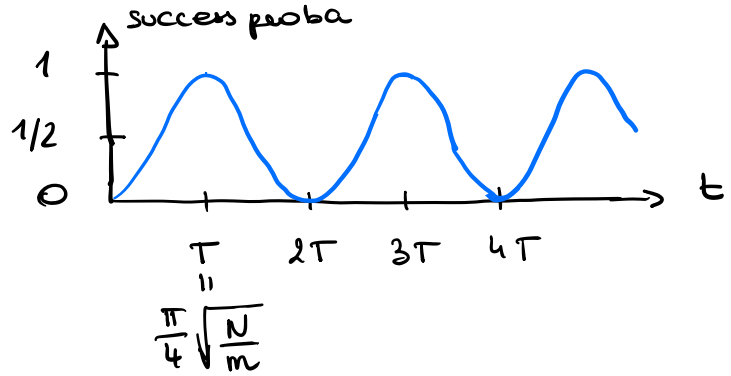
$\Rightarrow$ proba of error is $\sim \frac{m}{N} \ll 1$

(if $m \ll N$)

- The issue with Grover's algorithm is that if we apply the Grover iteration too many times, we will start rotating away from the winner state $|W\rangle$ $\Rightarrow$ really have to apply $T \approx \frac{\pi}{4}\sqrt{\frac{N}{m}}$ iterations.

But if don't know in advance how many solutions (how many $|w_i\rangle$) there are (m), then we don't know how many iterations we should apply $\rightarrow$ what can we do ?

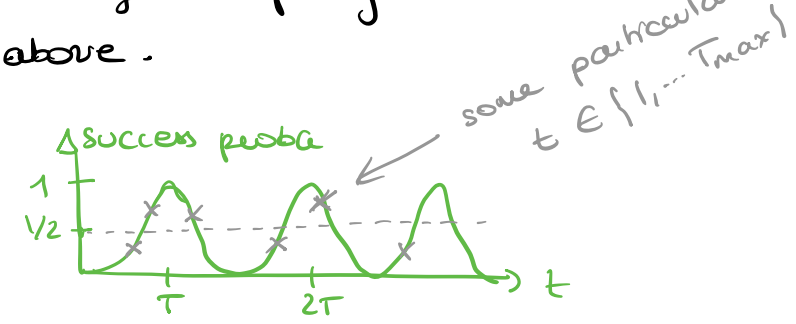
The success probability oscillates as a fct of the # of queries with period $2T \approx \frac{\pi}{2}\sqrt{\frac{N}{m}}$



In the case where there is one solution the period is the largest : $T = T_{max} = \frac{\pi}{4}\sqrt{N}$.

=> we can choose the number of queries by sampling from $\{1, 2, 3 \dots, T_{max}\}$.

=> we basically sample from the sinusoïde above.



=> Then if there is at least one solution, one is found with proba $\approx 1/2$ at each trial (because $\langle \sin^2 \theta \rangle = 1/2$)

If we repeat M times, sampling from a different random value of T each time, we will find a solution apart from (failure proba $\sim 2^{-M}$) .

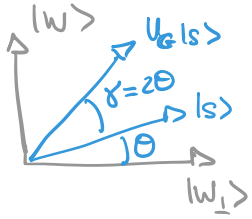
- Another possibility is to eshmate (count) the number of solutions in advance -
  ↳ "Quantum Counting" (see next tutorial)

The idea is that U_{Grover} has eigenvalues which depend on θ so we can use Quantum Phase Eshimation to evaluate $\theta \Rightarrow m$.

(since $\sin\theta = \sqrt{m/N}$)

Note: What happens if the number of solutions m is large? (not $m \ll N$)

One Grover iteration performs a rotation of angle $\gamma = 2\theta$ towards $|w\rangle$.



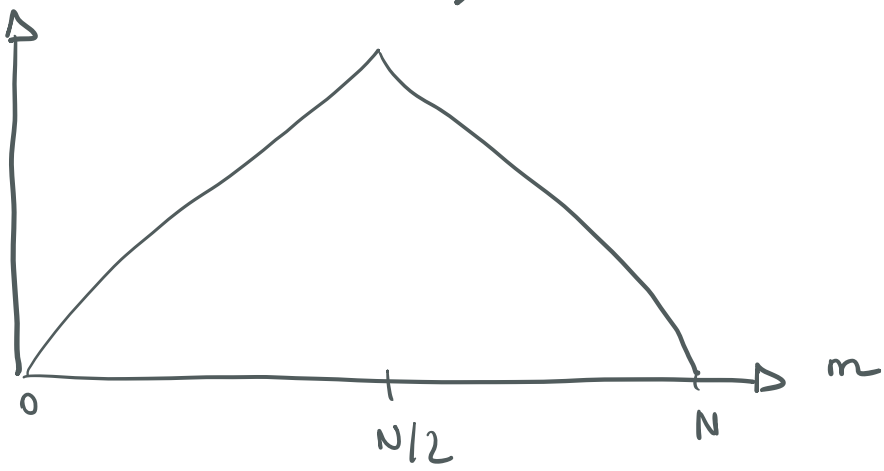
$$\Rightarrow \sin\gamma = \sin(2\theta)$$

$$= 2\cos\theta \sin\theta$$

$$= 2\sqrt{\frac{N-m}{N}}\sqrt{\frac{m}{N}}$$

$$= 2\frac{\sqrt{m(N-m)}}{N}$$

$$\gamma = \arcsin\left(2\frac{\sqrt{m(N-m)}}{N}\right)$$



For $m \in [0, \frac{N}{2}]$, the angle γ increases

$\Rightarrow$ the # of iterations T (≈ $\frac{\pi}{2\gamma} - \frac{1}{2}$) needed to solve the pb decreases

<u>However</u>, for $m \geq \frac{N}{2}$ the angle γ decreases as the # of solutions m increases.

$\Rightarrow$ the # of iterations needed increases with m for $m \geq N/2$.

$\Rightarrow$ weird - would expect it would become easier to find a solution as the # of solutions increases.

Two ways around this pb:

1) if we know in advance that $m \geq N/2$ - can solve the pb "classically": just pick a random state and check that it is a solution with the oracle. (proba 1/2 of success).

2) if we don't know in advance whether $m \geq \frac{N}{2}$: one can use "quantum counting" to first count the solution or one can use another procedure which consists in adding N extra states to the search space, that are not solutions to the problem.

Then the # of solutions $m < \frac{2N}{2}$ → size of the new search space.

This is done by adding on extra qubit $|q\rangle$

=> computational-basis states :

before :	after :
(initial search space)	(augmented search space)
$\|x\rangle$ = n-qubit states	$\|x\rangle \otimes \|q\rangle$
	= n+1 qubit states

An "augmented oracle" U_ω' is created which marks a state $|xq\rangle$ only if

↳ applies a $\ominus$ phase

$|x\rangle = |\omega_i\rangle$ = solution and $|q\rangle = |0\rangle$.

Now the number of iterations needed to find a solution is

$$T = \frac{\pi}{2\gamma} - \frac{1}{2}$$

with $\gamma = \ell\theta = \ell \sqrt{\frac{m}{2N}} = \sqrt{\frac{2m}{N}}$

$$\Rightarrow T = O\left(\sqrt{N/m}\right)$$

Note: Grover's algorithm is optimal

Grover's algo provides the fastest possible quantum search of an unsorted database if "time" is measured according to the number of queries of the oracle.

(See Nielsen & Chuang chap 6.6 for a derivation)

↳ it is proven that we need at least $O(\sqrt{N})$ calls to the oracle to achieve proba of success $\geq 1/2$ for finding a solution.

To do better one would have to exploit the structure of the pb.

Then the question is: do NP-complete pbs have some shared structure that could allow to solve them with an exponential quantum speed up ?

(in that case these NP-complete pbs would be in BQP $\rightarrow$ NP $\subseteq$ BQP)
so far no sign of such structure.