Grover's algorithm

Search in an unordered database

Example: phonebook, need to find a person from a phone number

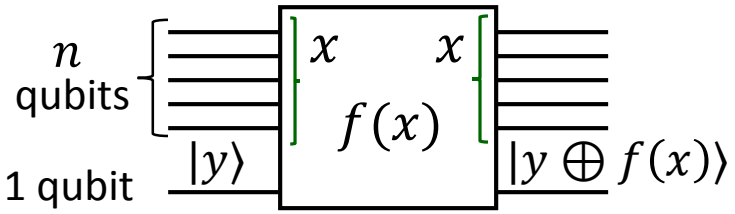
Actually, something else, like hard (e.g., NP-complete) problem

"Black box" $f(x) = \begin{cases} 0, & x \neq a \\ 1, & x = a \end{cases}$ $x = 0, 1, \ldots N-1$ $N \leq 2^n$

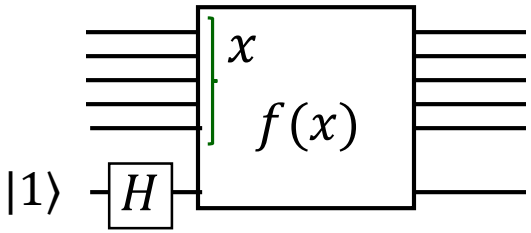
We want to find a

(so far only one a, later will generalize to several)

QC "oracle" (as usual)



Usual trick



$$|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \longrightarrow (-1)^{f(x)} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

So $|x\rangle \longrightarrow (-1)^{f(x)} |x\rangle$

Denote this operator $\hat{V}$: $\hat{V}|x\rangle = (-1)^{f(x)}|x\rangle$ (obviously unitary)

Operators $\hat{V}$ and $\hat{W}$

$$f(x) = \begin{cases} 1, & x = a \\ 0, & x \neq a \end{cases} \qquad \hat{V}|x\rangle = (-1)^{f(x)}|x\rangle$$

We can rewrite this unitary as $\hat{V} = \hat{1} - 2\underbrace{|a\rangle\langle a|}_{\text{projector onto } |a\rangle}$

Operator $\hat{V}$ changes sign of a component along $|a\rangle$
and does not change orthogonal component

Grover's algorithm uses operator $\hat{V}$ and also another operator $\hat{W}$

$$\hat{W} = 2\underbrace{|\Phi\rangle\langle\Phi|}_{\text{projector onto } |\Phi\rangle} - \hat{1}$$

$$|\Phi\rangle = H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$$

(equal superposition of all states)

Operator $\hat{W}$ does not change $|\Phi\rangle$, while all states orthogonal to $|\Phi\rangle$ change sign
(similar but opposite to what $\hat{V}$ is doing with $|a\rangle$)

Grover's algorithm

Quite simple algorithm. Goal: find a

$\ldots(\hat{W}\hat{V})(\hat{W}\hat{V})|\Phi\rangle$ Apply $\text{int}\,[(\pi/4)\sqrt{N}]$ times, measure.

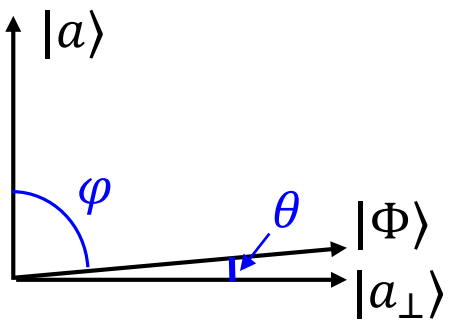
Will get a with high probability

$\hat{V} = \hat{1} - 2|a\rangle\langle a|$ $\quad \hat{W} = 2|\Phi\rangle\langle\Phi| - \hat{1}$ $\quad |\Phi\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$

Both $\hat{W}$ and $\hat{V}$ keep a vector in a plane spanned by $|a\rangle$ and $|\Phi\rangle$; moreover, components along by $|a\rangle$ and $|\Phi\rangle$ remain real (if initially real, yes).

$\Rightarrow$ Sufficient to consider only this plane (2D subspace of the Hilbert space)

$|a\rangle$ and $|\Phi\rangle$ are almost orthogonal to each other, but not exactly



$$\cos\varphi = \langle a|\Phi\rangle = \frac{1}{\sqrt{2^n}} = \frac{1}{\sqrt{N}}$$

Choose $N = 2^n$. If $N < 2^n$, then add entries on list

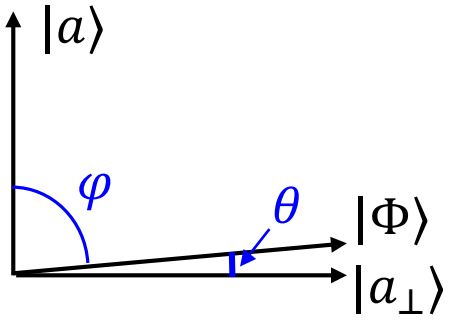
Introduce $|a_\perp\rangle$, $|a_\perp\rangle = \frac{1}{\sqrt{N-1}}\sum_{x\neq a}|x\rangle$

Then $\sin\theta = \cos\varphi = \frac{1}{\sqrt{N}} \quad \Rightarrow \quad \theta \approx \frac{1}{\sqrt{N}}$

Grover's algorithm: state evolution

$\left(\hat{W}\hat{V}\right)^k|\Phi\rangle$ Apply $k = \text{int}\,[(\pi/4)\sqrt{N}]$ operations $\hat{W}\hat{V}$, measure.

$\hat{V} = \hat{1} - 2|a\rangle\langle a|$ $\quad \hat{W} = 2|\Phi\rangle\langle\Phi| - \hat{1}$ $\quad |\Phi\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$



$\hat{V}$ is mirror-reflection about $|a_\perp\rangle$
(changes sign of a component along $|a\rangle$)

$\hat{W}$ is mirror-reflection about $|\Phi\rangle$

$\hat{W}\hat{V}$ is composition of two reflections $\Rightarrow$ rotation by 2θ

$\left(\hat{W}\hat{V}\right)^k$ is rotation by $2k\theta$. Initial state $|\Phi\rangle$ will become very close to $|a\rangle$ for

$$k = \frac{\pi/2 - \theta}{2\theta} \approx \frac{\pi}{4\theta} \approx \frac{\pi}{4}\sqrt{N}$$

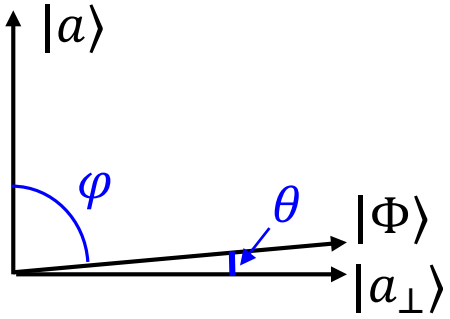
This explains the algorithm: start with $|\Phi\rangle$, apply $\text{int}\,[(\pi/4)\sqrt{N}]$ operations $\hat{W}\hat{V}$, measure

Probability of obtaining a is very close to 100%: $p_{\text{success}} \geq \cos^2\theta \approx 1 - 1/N$.

($\theta \approx 1/\sqrt{N}$, so $\cos\theta \approx 1 - 1/2N$)

Grover's algorithm: needed accuracy in k

$\left(\hat{W}\hat{V}\right)^k|\Phi\rangle$ Apply $k = \text{int}\,[(\pi/4)\sqrt{N}]$ operations $\hat{W}\hat{V}$, measure.



$$p_{\text{success}} \geq \cos^2\theta \approx 1 - 1/N.$$

Still OK if miss by 100 steps, then $\cos^2\theta \to \cos^2(200\,\theta)$

Even OK to use any $k \sim \sqrt{N}$, then probability of success is $\sim 50\%$.
Since easy to check, we can repeat the procedure until we find a.

Very robust procedure

Grover's algorithm changes search time from $\sim N$ to $\sim \sqrt{N}$
(not exponential, but still significant)

It was proven that Grover's algorithm is optimal in the sense that it is impossible to do the search with less than $\sim \sqrt{N}$ queries of the oracle in the quantum case

Grover's algorithm: generalization to several "solutions"

$$f(x)=\begin{cases}1, & x=a_1,a_2,..a_m\\ 0, & \text{otherwise}\end{cases}$$

Assume we know m and want to find a_i

The same procedure, just smaller number of operations $\hat{W}\hat{V}$

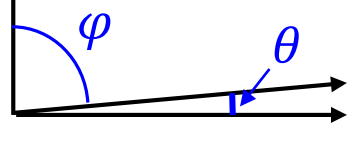
QC oracle: $|x\rangle \to (-1)^{f(x)}|x\rangle$ (still use $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ in the output register)

$$\hat{V}|x\rangle=(-1)^{f(x)}|x\rangle \qquad \text{Define} \quad |\Psi\rangle \equiv \frac{1}{\sqrt{m}}\sum_{i=1}^{m}|a_i\rangle$$

Now $\hat{V}$ changes sign of all components along $a_1,a_2,..a_m$. However, if we start with $|\Phi\rangle$, then states $\left(\hat{W}\hat{V}\right)^k|\Phi\rangle$ are still within the plane spanned by $|\Phi\rangle$ and $|\Psi\rangle$

(since we started with equal amplitudes for a_i and $\hat{W}$ or $\hat{V}$ do not distinguish individual a_i, they always come equally weighted, i.e., as a sum)

$$|\Psi\rangle=\frac{|a_1\rangle+|a_2\rangle+...|a_m\rangle}{\sqrt{m}}$$



$$|\Psi_\perp\rangle=\frac{1}{\sqrt{N-m}}\sum_{x\neq a_i}|x\rangle \qquad \sin\theta=\langle\Phi|\Psi\rangle=\frac{\sqrt{m}}{\sqrt{N}}$$

So, the only difference is the angle

Need $k=\text{int}[(\pi/4)\sqrt{N/m}]$ applications of $\hat{W}\hat{V}$

Then measure and get random a_i

If need all a_i, repeat $\sim m$ times, then total $\sim\sqrt{mN}$ uses of $\hat{V}$

Grover's algorithm: generalization (cont.)

$$f(x) = \begin{cases} 1, & x = a_1, a_2, .. a_m \\ 0, & \text{otherwise} \end{cases}$$

What if is m is unknown?

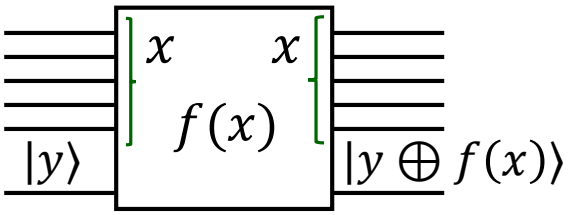
Still possible to find m first, and then find a_i as already discussed

Idea to find m: the function $\left(\hat{W}\hat{V}\right)^k |\Phi\rangle$ is almost periodic as a function of k, with period $\pi\sqrt{N/m}$, so we can use a period-finding (Kitaev's phase estimation) algorithm; complexity is also $\sim \sqrt{N}$, will need c-$\left(\hat{W}\hat{V}\right)^{2^j}$ operations (which require 2^j controlled queries of $\hat{V}$ -- cannot make more efficient; sufficient to use $\sim 2^{n/2} = \sqrt{N}$ times).

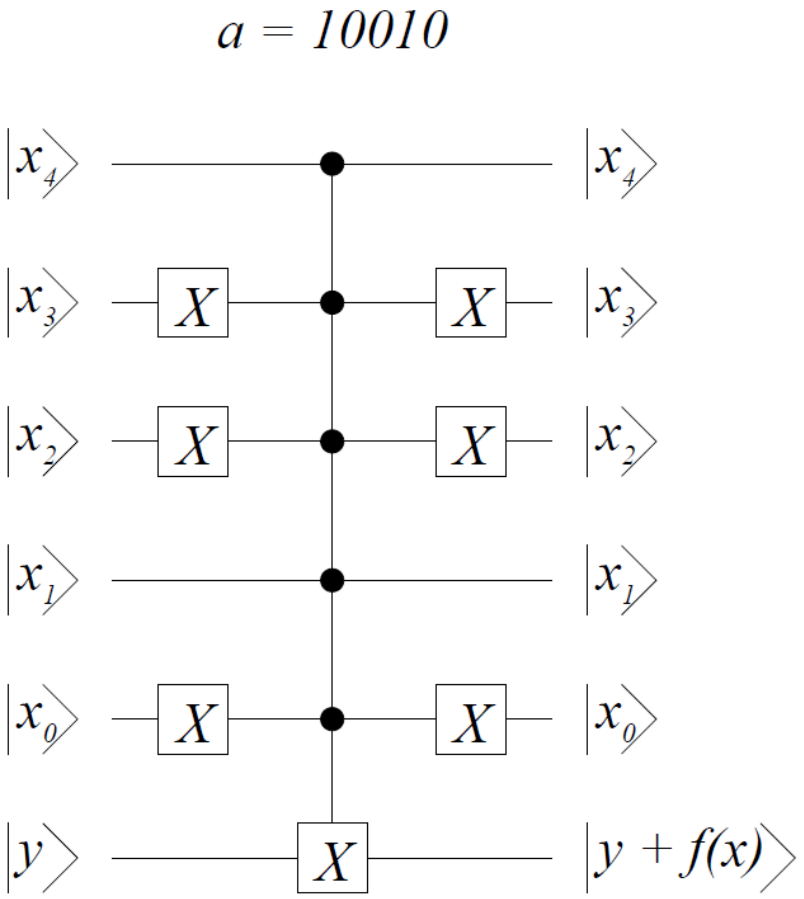
Construction of $\hat{V}$ and $\hat{W}$

$\hat{V}$ is an oracle, so we do not need to construct it.

Still, an example (for $m = 1$)



=



Construction of $\hat{V}$ and $\hat{W}$ (cont.)

Now construction for $\hat{W} = 2|\Phi\rangle\langle\Phi| - \hat{1}$

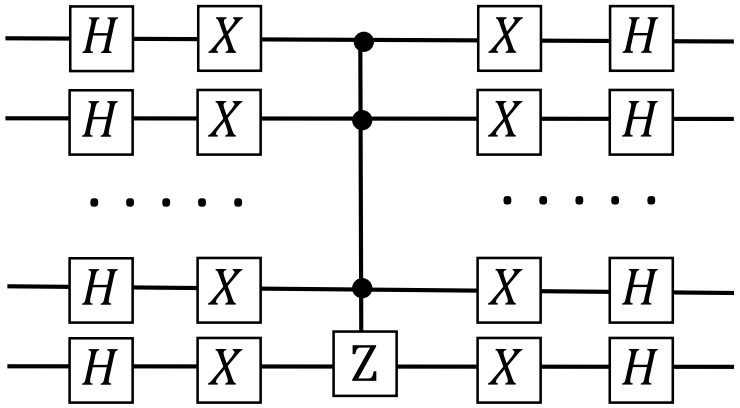
It is easier to construct $-\hat{W}$ (does not matter, since only an overall sign)

$$-\hat{W} = \hat{1} - 2|\Phi\rangle\langle\Phi| = \hat{1} - 2\underbrace{H^{\otimes n}|0_n\rangle}_{|\Phi\rangle}\underbrace{\langle 0_n|H^{\otimes n}}_{\langle\Phi|} =$$

$$= H^{\otimes n}\,(\underbrace{\hat{1} - 2\,|0_n\rangle\langle 0_n|}\,)\;H^{\otimes n}$$

applies $-\hat{1}$ instead of $\hat{1}$ when all zeros
Similar to controlled^{n-1}-Z, but works when all 0s instead of all 1s, so we need to exchange $0 \leftrightarrow 1$

$-\hat{W}$ =

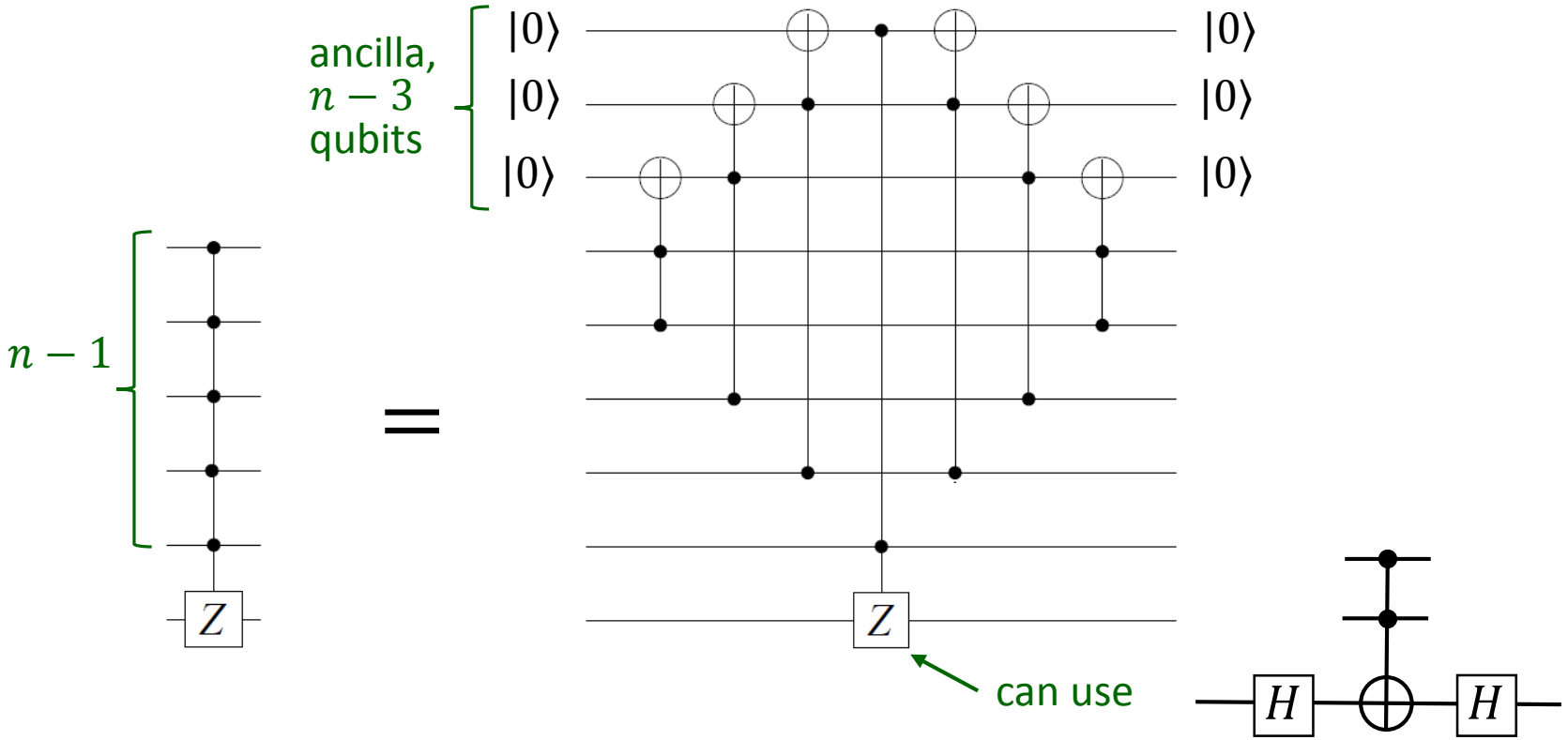


c^{n-1}-Z gate is symmetric, so the circuit is symmetric

Construction of $\hat{V}$ and $\hat{W}$ (cont.)

We still need to construct controlled^{n-1}-Z gate out of 1-qubit and 2-qubit gates

In principle (not the best) it can be constructed in the following way



since $HXH = Z$
and $H^2 = \hat{1}$

So, c^{n-1}-$Z \rightarrow$ Toffoli $\rightarrow$ CNOTs and 1-qubit gates

Realization of an arbitrary unitary operation

Theorem An arbitrary *n*-qubit unitary operation can be realized using CNOTs and 1-qubit unitaries

Long proof by explicit construction. Steps in the proof:

1) an arbitrary unitary can be realized by control^{n-1}-U gates and state permutations (using CNOTs)
2) control^{n-1}-U gates can be realized by Toffoli and control-U gates
3) Toffoli and control-U gates can be realized by CNOTs and 1-qubit gates

However, this construction may require exponential number of gates

An arbitrary n-qubit unitary is characterized by 2^{2n}-1 real parameters (matrix $2^n \times 2^n$, complex numbers, but 2^{2n} equations for unitarity, so 2^{2n} remaining parameters, also overall phase)

But each gate is characterized only by a few parameters (3 parameters for 1-qubit gate, no parameters for CNOT)

Therefore $\sim 2^{2n}$ gates are needed (actually even $\sim n\, 2^{2n}$)

So, having an efficient circuit (polynomial in n) is a luck

Efficient and inefficient QC algorithms

In general, n-qubit unitary requires $\sim 2^{2n}$ gates

So, having an efficient circuit (polynomial in n) is a luck

However, if it exists, then two good things:

1. Precision of the gates is not a big problem (required imprecision scales linearly with the number of gates)
2. It is sufficient to use a universal set of gates

Universal sets of quantum gates

Exact:

CNOT and all one-qubit gates

CNOT, H, and $\begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}$ (continuous φ)

Toffoli and all one-qubit gates

and so on . . .

Discrete universal sets of gates

Approximate:

"Standard set" (N-C): CNOT, H, $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$, $T = \sqrt{S} = "\pi/8"$

also: CNOT, H, $T = "\pi/8"$

also: CNOT can be replaced with c$-Z$ or any two-qubit entangling gate

often X, Y, Z are added (makes design easier)

often Toffoli is added (makes design easier)

Solovay-Kitaev theorem:

If we require inaccuracy less than ε for a desired unitary operation,
it can be realized using a universal set of gates,
$\|U_{\text{desired}} - U_1 U_2 \dots U_k\| < \varepsilon$,
with overhead complexity (number of gates) scaling as $\log^2(1/\varepsilon)$.

Still, some sets are not universal.

For example, the set CNOT, X, Y, Z, H is not sufficient (Clifford gates)

Gottesman-Knill theorem: QC circuits made of Clifford gates can be efficiently simulated on a classical computer. (Clifford group: normalizer of the Pauli group)