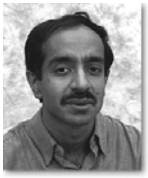
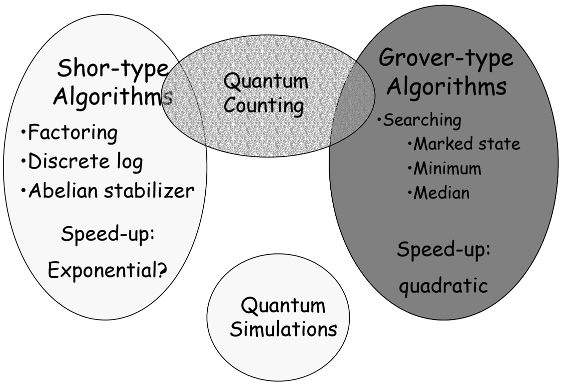
Grover's Algorithm



Lov K. Grover
Bell Labs

Grover
Sesame Street

Quantum Algorithms



Unsorted Database

- Example: Telephone Book

Find the name of the person with phone number: 3397 0454

- Very difficult task!
- If there are N entries in the phone book, it will take an average of N/2 queries to find the name

Unsorted Database

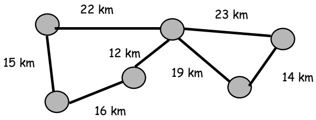
- Example: Telephone Book

Find the name of the person with phone number: 3397 0454

- This is not such a good example
  - There exists a more efficient solution
  - The search time is linear with respect to the size of the problem
  - That is because the number of possible inputs scales linearly with the size of the problem

Traveling salesman Problem

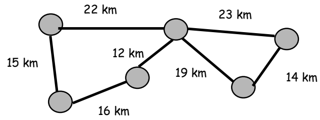
Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- What is the size of this problem?
  - The network might be directed (It could take longer to get from A to B than from B to A)
  - Each city could be connected to every other city
  - Therefore if there are c cities, there could be c(c-1) edges
  - Each edge requires log k bits to store

Traveling salesman Problem

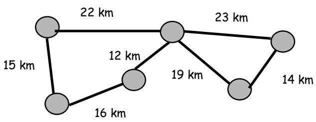
Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- The size of the problem is $O(c^2 \log k)$,
- How many possible tours are there?
  - Equal to the number of permutations of cities
  - There are c! possible tours
  - $c! \sim O(\exp(c \log c))$
- Therefore the number of possible inputs scales exponentially with the size of the problem

Traveling salesman Problem

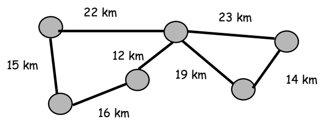
Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- Suppose there is exactly one solution
- We could try to solve the problem by choosing a tour and testing if it is less than k
- It would take on average c!/2 attempts to find the solution
- Grover's algorithm allows us to find a solution by using only $O(\sqrt{c!})$ attempts

Traveling salesman Problem

Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- $O(\sqrt{c!})$ is still exponential
- We haven't made the problem "tractable"
- Suppose there are 10 billion permutations
- Grover's algorithm would only require one hundred thousand queries
- Classically we would require an average of 5 billion queries

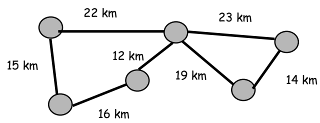
The Quantum Oracle

oracle [or'a-kl] *n* a medium or agency, especially in ancient Greece, of divine revelation; a person of great wisdom; a wise utterance

- Not an all-knowing device
- Simple a device which can efficiently check whether a given solution is correct
- A witness
- Telephone book:
  - Is Jones the name of the person with phone number 3397 0454?

The Quantum Oracle

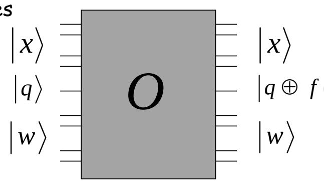
Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- We could write a computer program to check whether a tour is a valid solution
- We could make it reversible
- Therefore, we could implement it on a quantum computer

The Quantum Oracle

- Imagine that the bit string, x, represents a tour of the cities



$$|x\rangle|q\rangle|w\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle|w\rangle$$

$$f(x) = \begin{cases} 1, & \text{if } x \text{ is a solution} \\ 0, & \text{if } x \text{ is not a solution} \end{cases}$$

The Quantum Oracle

- The quantum oracle bit-flips the oracle qubit if the input is a valid solution
- The inner workings of the oracle are by no means `magical'
- In the traveling salesman example, the oracle qubit would be flipped if x encoded a tour of the cities with a distance less than k.
- By abstracting the problem using an oracle, we can forget about the specific problem we are trying to solve

The Quantum Oracle

$$|x\rangle|q\rangle|w\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle|w\rangle$$

- The work qubits are returned to their initial state, so we will ignore them

$$|x\rangle|q\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle$$

- Suppose the oracle qubit is initially in the state

$$|q\rangle = \tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle$$

The Quantum Oracle

- If x is not a solution, the oracle does nothing

$$|x\rangle(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle) \xrightarrow{O} |x\rangle(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle)$$

- If x is a solution, the oracle qubit is flipped

$$|x\rangle(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle) \xrightarrow{O} -|x\rangle(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle)$$

- We can write both of these as

$$|x\rangle(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle) \xrightarrow{O} (-1)^{f(x)}|x\rangle(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle)$$

- Or simply as

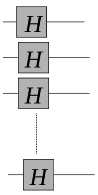
$$|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle$$

Hadamard gate

- Remembering the Hadamard gate,

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- In a slight abuse of notation,



$$= H^{\otimes n}$$

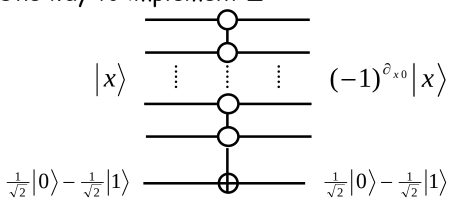
Will written simply as H

$$|\psi\rangle \equiv H|0\rangle = \frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle$$

Zero state phase shift

$$|x\rangle \xrightarrow{Z} (-1)^{\partial_{x0}}|x\rangle$$

- One way to implement Z:



- Flips the oracle qubit iff $|x\rangle = |0\rangle$

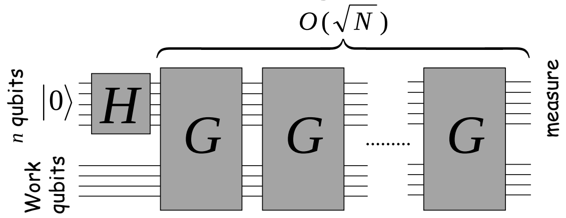
Zero state phase shift

- The matrix representation of Z

$$Z = \begin{bmatrix} -1 & 0 & 0 & & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ & \vdots & & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Z = 2|0\rangle\langle 0| - I$$

Grover's Algorithm



$$G = HZHO$$

(1) Apply the oracle
(2) Apply the Hadamards
(3) Apply the zero state phase shift
(4) Apply the Hadamards

Grover Iterate

$$G = HZHO$$

- O: inverts the solution states
- HZH: invert all states about the mean

$$HZH$$
$$H(2|0\rangle\langle 0| - I)H$$
$$2H|0\rangle\langle 0|H - HIH$$
$$2|\psi\rangle\langle\psi| - HIH$$
$$2|\psi\rangle\langle\psi| - I$$

Grover Iterate

$$HZH = 2|\psi\rangle\langle\psi| - I$$

$$(|\psi\rangle\langle\psi|)|\alpha\rangle = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & & 1 \\ & \vdots & & \ddots & \\ 1 & 1 & 1 & & 1 \end{bmatrix}\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} \frac{\sum \alpha_n}{N} \\ \frac{\sum \alpha_n}{N} \\ \frac{\sum \alpha_n}{N} \\ \vdots \\ \frac{\sum \alpha_n}{N} \end{bmatrix}$$

Grover Iterate

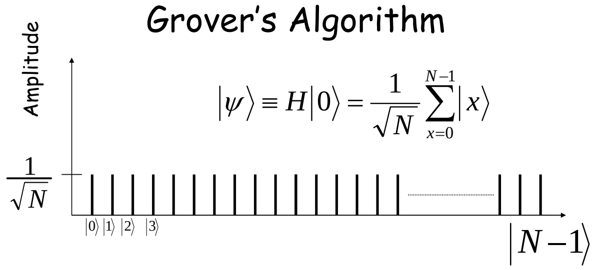
$$HZH = 2|\psi\rangle\langle\psi| - I$$

$$HZH|\alpha\rangle = (2|\psi\rangle\langle\psi| - I)\sum_n \alpha_n |n\rangle$$
$$= \sum_n 2\overline{\alpha} - \alpha_n |n\rangle$$

HZH: invert all states about the mean

Grover's Algorithm

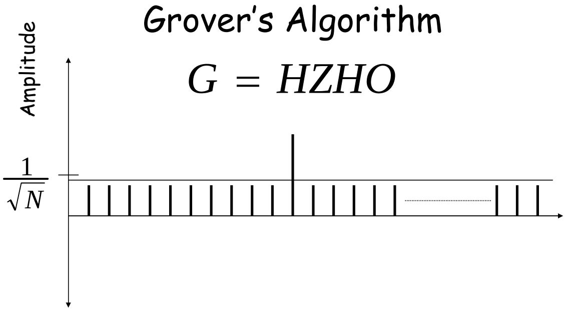
$$|\psi\rangle \equiv H|0\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$$



- The probability of measuring the marked state, $P(m) = \frac{1}{N} = \frac{1}{2^n}$

Grover's Algorithm

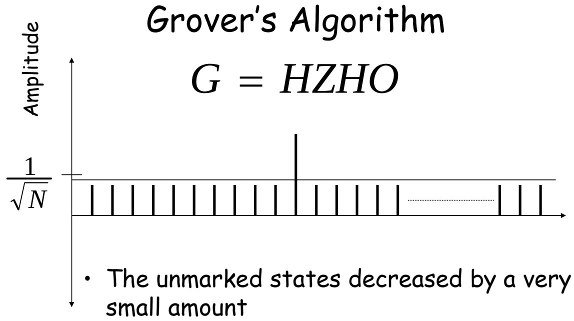
$$G = HZHO$$



- O: invert the solution state
- HZH: invert all states about the mean

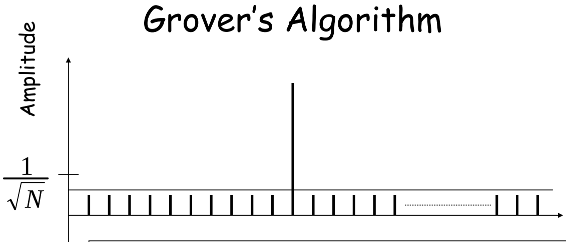
Grover's Algorithm

$$G = HZHO$$



- The unmarked states decreased by a very small amount
- The marked state increased by $O(\frac{1}{\sqrt{N}})$
- To get P(m) = O(1), we need to apply the Grover iterate $O(\sqrt{N})$ times.

Grover's Algorithm



Exercise: Suppose we apply Grover's algorithm to a 4 qubit register, assuming exactly one of the states is marked.

What is the probability of measuring the marked state after applying the Grover iterate 0,1,2,3 times?

How many Grover iterates do we need?

- Initial amplitudes of the marked and unmarked states:

$$m_0 = \frac{1}{\sqrt{N}} \qquad u_0 = \frac{1}{\sqrt{N}}$$

- After inverting the marked state, the average amplitude is

$$a_i = \frac{(N-1)u_{i-1} - m_{i-1}}{N}$$

- Completing the Grover iterate

$$m_i = 2a_i + m_{i-1}$$

$$u_i = 2a_i - u_{i-1}$$

How many Grover iterates do we need?

- Substituting a_i in gives

$$m_i = 2(1-\tfrac{1}{N})u_{i-1} + (1-\tfrac{2}{N})m_{i-1}$$

$$u_i = 2(1-\tfrac{2}{N})u_{i-1} - \tfrac{2}{N}m_{i-1}$$

- We would like to find k, such that

$$|m_k| \geq \tfrac{1}{\sqrt{2}} \quad \text{so that} \quad P(m_k) \geq \tfrac{1}{2}$$

- Note that

$$\begin{bmatrix} m_k \\ u_k \end{bmatrix} = \begin{bmatrix} 1-\frac{2}{N} & 2-\frac{2}{N} \\ -\frac{2}{N} & 1-\frac{2}{N} \end{bmatrix}^k \begin{bmatrix} \frac{1}{\sqrt{N}} \\ \frac{1}{\sqrt{N}} \end{bmatrix}$$

$$\boxed{k = \left\lceil \frac{\pi\sqrt{N}}{4} \right\rceil}$$

Generalizations of Grover's Algorithm

- What if we have more than one marked state?

$$k = \left\lceil \frac{4}{\pi}\sqrt{\frac{N}{M}} \right\rceil$$

  - M: Number of solutions
  - M < N/2

- What if we use a different initial state?
  - The algorithm still works

- What if we alter the Grover iterate?
  - The algorithm still works

Optimality of search algorithm

- Classically, if all you can do is ask questions of the oracle
  - The best you can do is $O(N)$

- Quantumly, if all you can do is ask questions of the oracle
  - The best you can do is $O(\sqrt{N})$

Summary

- Used to solve a problem that you don't know much about:
  - Unsorted databases
  - NP-complete problems
- Problems remain intractable
- Square-root speed-up