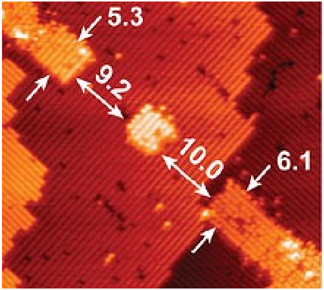
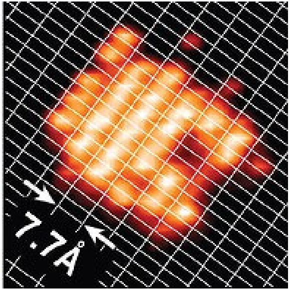
Introduction to Quantum Computing
Part II

Emma Strubell

http://cs.umaine.edu/~ema/quantum_tutorial.pdf

April 13, 2011





Overview

Grover's Algorithm

- Quantum search
- How it works
- A worked example

Simon's algorithm

- Period-finding
- How it works
- An example

Outline

Grover's Algorithm

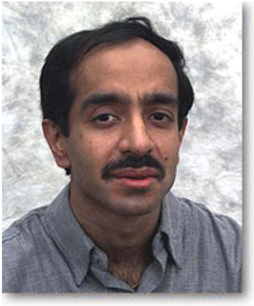
- Quantum search
- How it works
- A worked example

Simon's algorithm

- Period-finding
- How it works
- An example

Quantum search: quadratic speedup

- Performs a search over an unordered set of $N = 2^n$ items to find the unique element that satisfies some condition
- Best classical algorithm requires $O(N)$ time
- Grover's algorithm performs the search in only $O(\sqrt{N})$ operations, a quadratic speedup
- If the algorithm were to run in a finite power of $O(\lg N)$ steps, then it would provide an algorithm in BQP for NP-complete problems
- But no, Grover's algorithm is optimal for a quantum computer



Outline

Grover's Algorithm

- Quantum search
- How it works
- A worked example

Simon's algorithm

- Period-finding
- How it works
- An example

Step 1: Attain equal superposition

- Begin with a quantum register of n qubits, where n is the number of qubits necessary to represent the search space of size $2^n = N$, all initialized to $|0\rangle$:

$$|0\rangle^{\otimes n} = |0\rangle \tag{1}$$

- First step: put the system into an equal superposition of states, achieved by applying the Hadamard transform $H^{\otimes n}$

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

- Requires $\Theta(\lg N) = \Theta(\lg 2^n) = \Theta(n)$ operations, n applications of the elementary Hadamard gate:

Amplitude amplification: the Grover iteration

- Next series of transformations often referred to as the *Grover iteration*
- Bulk of the algorithm
- Performs *amplitude amplification*
  - Selective shifting of the phase of one state of a quantum system, one that satisfies some condition, at each iteration
  - Performing a phase shift of π is equivalent to multiplying the amplitude of that state by -1: amplitude for that state changes, but the probability remains the same
  - Subsequent transformations take advantage of difference in amplitude to single state of differing phase, ultimately increasing the probability of the system being in that state
- In order to achieve optimal probability that the state ultimately observed is the correct one, want overall rotation of the phase to be $\frac{\pi}{4}$ radians, which will occur on average after $\frac{\pi}{4}\sqrt{2^n}$ iterations
- The Grover iteration will be repeated $\frac{\pi}{4}\sqrt{2^n}$ times

The Grover iteration: an oracle query

- First step in Grover iteration is a call to a *quantum oracle*, $\mathcal{O}$, that will modify the system depending on whether it is in the configuration we are searching for
- An oracle is basically a black-box function, and this quantum oracle is a quantum black-box, meaning it can observe and modify the system without collapsing it to a classical state
- If the system is indeed in the correct state, then the oracle will rotate the phase by π radians, otherwise it will do nothing
- In this way it marks the correct state for further modification by subsequent operations
- The oracle's effect on $|x\rangle$ may be written simply:

$$|x\rangle \xrightarrow{\mathcal{O}} (-1)^{f(x)} |x\rangle \tag{3}$$

  Where $f(x) = 1$ if x is the correct state, and $f(x) = 0$ otherwise
- The exact implementation of $f(x)$ is dependent on the particular search problem

The Grover iteration: diffusion transform

- Grover refers to the next part of the iteration as the *diffusion transform*
- Performs *inversion about the average*, transforming the amplitude of each state so that it is as far above the average as it was below the average prior to the transformation
- Consists of another application of the Hadamard transform $H^{\otimes n}$, followed by a conditional phase shift that shifts every state except $|0\rangle$ by -1, followed by yet another Hadamard transform
- The conditional phase shift can be represented by the unitary operator $2|0\rangle\langle 0| - I$:

$$[2|0\rangle\langle 0| - I]|0\rangle = 2|0\rangle\langle 0|0\rangle - I = |0\rangle \tag{4a}$$

$$[2|0\rangle\langle 0| - I]|x\rangle = 2|0\rangle\langle 0|x\rangle - I = -|x\rangle \tag{4b}$$

The Grover iteration: bringing it all together

- The entire diffusion transform, using the notation $|\psi\rangle$ from equation 2, can be written:

$$H^{\otimes n} \left[2\left|0\right\rangle\left\langle 0\right| - I\right] H^{\otimes n} = 2H^{\otimes n}\left|0\right\rangle\left\langle 0\right| H^{\otimes n} - I = 2\left|\psi\right\rangle\left\langle\psi\right| - I \quad (5)$$

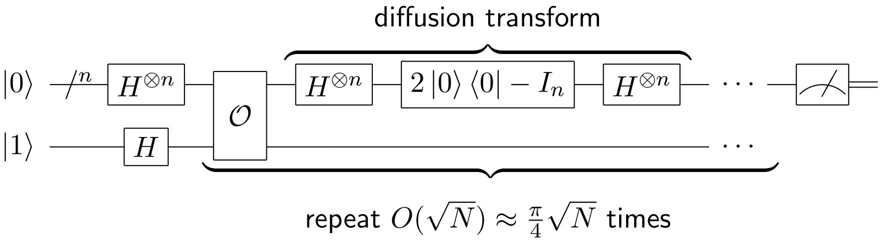
  And the entire Grover iteration:

$$\left[2\left|\psi\right\rangle\left\langle\psi\right| - I\right]\mathcal{O} \quad (6)$$

- The exact runtime of the oracle depends on the specific problem and implementation, so a call to $\mathcal{O}$ is viewed as one elementary operation
- Total runtime of a single Grover iteration is $O(n)$:
  - $O(2n)$ from the two Hadamard transforms
  - $O(n)$ gates to perform the conditional phase shift
- The runtime of Grover's entire algorithm, performing $O(\sqrt{N}) = O(\sqrt{2^n}) = O(2^{\frac{n}{2}})$ iterations each requiring $O(n)$ gates, is $O(2^{\frac{n}{2}})$.

Circuit diagram overview

- Once the Grover iteration has been performed $O(\sqrt{N})$ times, a classical measurement is performed to determine the result, which will be correct with probability $O(1)$



Outline

Grover's algorithm on 3 qubits

- Consider a system consisting of $N = 8 = 2^3$ states
- The state we are searching for, x_0, is represented by the bit string 011
- To describe this system, $n = 3$ qubits are required:

$$\begin{aligned} |x\rangle = \alpha_0 |000\rangle + \alpha_1 |001\rangle + \alpha_2 |010\rangle + \alpha_3 |011\rangle \\ + \alpha_4 |100\rangle + \alpha_5 |101\rangle + \alpha_6 |110\rangle + \alpha_7 |111\rangle \end{aligned}$$

  where α_i is the amplitude of the state $|i\rangle$
- Grover's algorithm begins with a system initialized to 0:

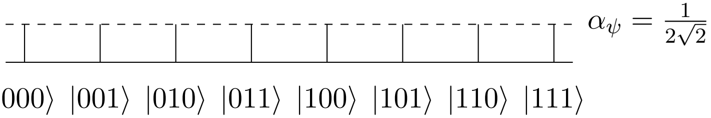
$$1\,|000\rangle$$

Attain equal superposition

- apply the Hadamard transformation to obtain equal amplitudes associated with each state of $1/\sqrt{N} = 1/\sqrt{8} = 1/2\sqrt{2}$, and thus also equal probability of being in any of the 8 possible states:

$$\begin{aligned} H^3 |000\rangle &= \frac{1}{2\sqrt{2}} |000\rangle + \frac{1}{2\sqrt{2}} |001\rangle + \ldots + \frac{1}{2\sqrt{2}} |111\rangle \\ &= \frac{1}{2\sqrt{2}} \sum_{x=0}^{7} |x\rangle \\ &= |\psi\rangle \end{aligned}$$

- Geometrically:

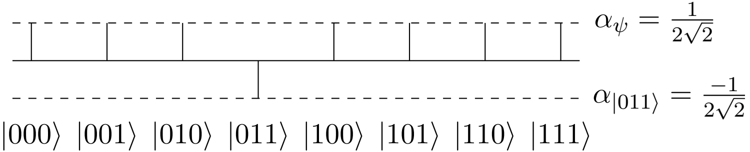


Two Grover iterations: the first Hadamard

- It is optimal to perform 2 Grover iterations:
  $\frac{\pi}{4}\sqrt{8} = \frac{2\pi}{4}\sqrt{2} = \frac{\pi}{2}\sqrt{2} \approx 2.22$ rounds to 2 iterations.
- At each iteration, the first step is to query $\mathcal{O}$, then perform inversion about the average, the diffusion transform.
- The oracle query will negate the amplitude of the state $|x_0\rangle$, in this case $|011\rangle$, giving the configuration:

$$|x\rangle = \frac{1}{2\sqrt{2}}\,|000\rangle + \frac{1}{2\sqrt{2}}\,|001\rangle + \frac{1}{2\sqrt{2}}\,|010\rangle - \frac{1}{2\sqrt{2}}\,|011\rangle + \ldots + \frac{1}{2\sqrt{2}}\,|111\rangle$$

- With geometric representation:



Diffusion transform

- Now perform the diffusion transform $2\left|\psi\right\rangle\left\langle\psi\right| - I$, which will increase the amplitudes by their difference from the average, decreasing if the difference is negative:

$$
\begin{aligned}
&[2\left|\psi\right\rangle\left\langle\psi\right| - I]\left|x\right\rangle \\
&= [2\left|\psi\right\rangle\left\langle\psi\right| - I]\left[\left|\psi\right\rangle - \frac{2}{2\sqrt{2}}\left|011\right\rangle\right] \\
&= 2\left|\psi\right\rangle\left\langle\psi|\psi\right\rangle - \left|\psi\right\rangle - \frac{2}{\sqrt{2}}\left|\psi\right\rangle\left\langle\psi|011\right\rangle + \frac{1}{\sqrt{2}}\left|011\right\rangle
\end{aligned}
$$

- Note that $\left\langle\psi|\psi\right\rangle = 8\frac{1}{2\sqrt{2}}\left[\frac{1}{2\sqrt{2}}\right] = 1$
- Since $\left|011\right\rangle$ is one of the basis vectors, we can use the identity $\left\langle\psi|011\right\rangle = \left\langle 011|\psi\right\rangle = \frac{1}{2\sqrt{2}}$

Diffusion transform continued

- Final result of the diffusion transform:

$$
\begin{aligned}
&= 2\,|\psi\rangle - |\psi\rangle - \frac{2}{\sqrt{2}}\left(\frac{1}{2\sqrt{2}}\right)|\psi\rangle + \frac{1}{\sqrt{2}}\,|011\rangle \\
&= |\psi\rangle - \frac{1}{2}\,|\psi\rangle + \frac{1}{\sqrt{2}}\,|011\rangle \\
&= \frac{1}{2}\,|\psi\rangle + \frac{1}{\sqrt{2}}\,|011\rangle
\end{aligned}
$$

- Substituting for $|\psi\rangle$ gives:

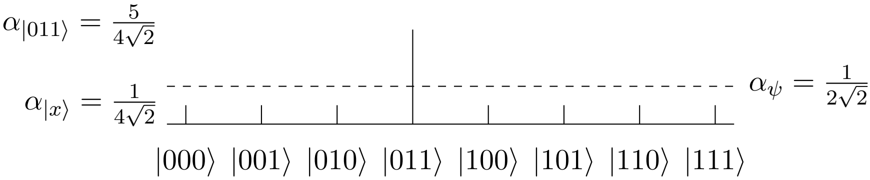
$$
\begin{aligned}
&=\frac{1}{2}\left[\frac{1}{2\sqrt{2}}\sum_{x=0}^{7}|x\rangle\right] + \frac{1}{\sqrt{2}}\,|011\rangle \\
&=\frac{1}{4\sqrt{2}}\sum_{\substack{x=0\\x\neq 3}}^{7}|x\rangle + \frac{5}{4\sqrt{2}}\,|011\rangle
\end{aligned}
$$

Geometric result of the diffusion transform

- Can also be written:

$$|x\rangle = \frac{1}{4\sqrt{2}}|000\rangle + \frac{1}{4\sqrt{2}}|001\rangle + \frac{1}{4\sqrt{2}}|010\rangle + \frac{5}{4\sqrt{2}}|011\rangle + \ldots + \frac{1}{4\sqrt{2}}|111\rangle$$

- Geometric representation:



The second Grover iteration

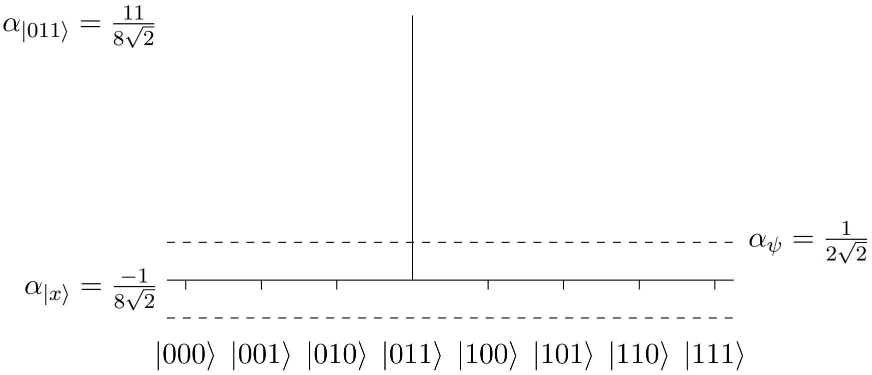
- I will spare you the details, as they are very similar. Result:

$$[2|\psi\rangle\langle\psi| - I]\left[\frac{1}{2}|\psi\rangle - \frac{3}{2\sqrt{2}}|011\rangle\right] = -\frac{1}{8\sqrt{2}}\sum_{\substack{x=0\\x\neq 3}}^{7}|x\rangle + \frac{11}{8\sqrt{2}}|011\rangle$$

- Longer format:

$$|x\rangle = -\frac{1}{8\sqrt{2}}|000\rangle - \frac{1}{8\sqrt{2}}|001\rangle - \frac{1}{8\sqrt{2}}|010\rangle + \frac{11}{8\sqrt{2}}|011\rangle - \ldots - \frac{1}{8\sqrt{2}}|111\rangle \tag{7}$$

Geometrically, the success of the algorithm is clear



Final answer

- When the system is observed, the probability that the state representative of the corrct solution, $|011\rangle$, will be measured is $|\frac{11}{8\sqrt{2}}|^2 = 121/128 \approx 94.5\%$
- The probability of finding an incorrect state is $|\frac{-\sqrt{7}}{8\sqrt{2}}|^2 = 7/128 \approx 5.5\%$
- Grover's algorithm is more than 17 times more likely to give the correct answer than an incorrect one with an input size of $N = 8$
- Error only decreases as the input size increases
- Although Grover's algorithm is probabilistic, the error truly becomes negligible as N grows large.

Outline

Simon's problem

- Simon's problem is, given a function

$$f : \{0,1\}^n \rightarrow \{0,1\}^n$$

known to be invariant under some n-bit XOR mask a, determine a

- In other words, determine a given:

$$f(x) = f(y) \longleftrightarrow x \oplus y \in \{0^n, a\}$$

- One of the first problems for which a quantum algorithm was found to provide exponential speedup over any classical algorithm
- Best classical algorithms, including probabilistic ones, require an exponential $\Omega(2^{n/2})$ queries to the black-box function in order to determine a
- Simon's quantum algorithm solves this problem in polynomial time, performing an optimal $O(n)$ queries

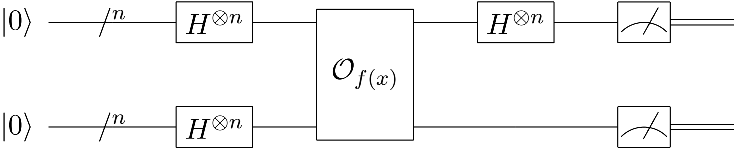
Period-finding, like Shor

- Simon's algorithm and Shor's prime factorization algorithm solve a similar problem: given a function f, find the period a of that function
- While Simon's problem uses XOR to define the period, Shor's uses binary addition as the constraint on f
- These problems are more restricted cases of what is known as the hidden subgroup problem, which corresponds to a number of important problems in computer science
- Any formulation of the Abelian hidden subgroup problem can be solved by a quantum computer requiring a number of operations logarithmic in the size of the group
- The more general hidden subgroup problem is harder to solve:
  - Analogous to the graph isomorphism problem, some shortest vector problems in lattices,
  - Currently no polynomial-time algorithms have been devised to solve this problem
  - Would be a breakthrough in quantum computing similar to Shor's discovery

Outline

More Hadamards, more oracle

- Overview of Simon's algorithm by circuit diagram
- Hadamard gates are important



Equal superposition, again, and then an oracle query

- Given a function acting on n-bit strings, Simon's algorithm begins by initializing two n-bit registers to 0:

$$|0\rangle^{\otimes n} |0\rangle^{\otimes n}$$

- Then applying the Hadamard transform to the first register to attain an equal superposition of states:

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

- Next, $f(x)$ is queried on both the registers
- The oracle is implemented as a unitary operation that performs the transformation $\mathcal{O}_{f(x)} |x\rangle |y\rangle = |x\rangle |f(x) \oplus y\rangle$:

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$$

Mid-algorithm measurement?!

- Now the second register is measured
- Two possible cases to consider in determining the impact of that measurment on the first register
  - XOR mask $a = 0^n$
  - $a = \{0,1\}^n$
- If $a = 0^n$, then f is injective: each value of x corresponds to a unique value $f(x)$
- This means that the first register remains in an equal superposition; Regardless of the measured value of $f(x)$, x could be any bit string in $\{0,1\}^n$ with equal probability
- If $a = \{0,1\}^n$, measuring the second register determines a concrete value of $f(x)$, call it $f(z)$, which limits the possible values of the first register
- Two possible values of x such that $f(x) = f(z)$: z and $z \oplus a$:

$$\frac{1}{\sqrt{2}}|z\rangle + \frac{1}{\sqrt{2}}|z \oplus a\rangle$$

Extracting information about a: Hadamard, of course

- Since there will be no more operations on the second register, further calculations will focus only on the first register.
- The next step is to isolate the information about a that is now stored in the first register
- This can be done by applying the Hadamard transform again
- The Hadamard transform may be defined using the bitwise dot product $x \cdot y$ as:

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

- Using this notation, the result of applying a second Hadamard operation is:

$$
\begin{aligned}
& H^{\otimes n}\left[\frac{1}{\sqrt{2}}\left|z\right\rangle + \frac{1}{\sqrt{2}}\left|z \oplus a\right\rangle\right] \\
=& \frac{1}{\sqrt{2}} H^{\otimes n}\left|z\right\rangle + \frac{1}{\sqrt{2}} H^{\otimes n}\left|z \oplus a\right\rangle \\
=& \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{z \cdot y}\left|y\right\rangle\right] + \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{(z \oplus a) \cdot y}\left|y\right\rangle\right] \\
=& \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} \left[(-1)^{z \cdot y} + (-1)^{(z \oplus a) \cdot y}\right]\left|y\right\rangle \\
=& \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} \left[(-1)^{z \cdot y} + (-1)^{(z \cdot y) \oplus (a \cdot y)}\right]\left|y\right\rangle \\
=& \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} (-1)^{z \cdot y}\left[1 + (-1)^{a \cdot y}\right]\left|y\right\rangle
\end{aligned}
$$

Final measurement

- Now the value of the first register is measured
- In the degenerate case where $a = 0^n$ (f is injective), a string will be produced from $\{0,1\}^n$ with uniform distribution
- In the case where $x \oplus y \neq 0^n$, notice that either $a \cdot y = 0$ or $a \cdot y = 1$. If $a \cdot y = 1$, which gives:

$$\begin{aligned}\frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} (-1)^{z \cdot y} \left[1 + (-1)^1\right] |y\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} (-1)^{z \cdot y} \left[0\right] |y\rangle \\ &= 0 \, |y\rangle\end{aligned}$$

- The amplitude, and thus probability, that a value of y such that $a \cdot y = 1$ is equal to 0, and so such a y will never be measured.

More on the final measurement

- Knowing that it will always be true that $a \cdot y = 0$, the equation can yet again be simplified:

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} (-1)^{z \cdot y} \left[1 + (-1)^0\right] |y\rangle = \frac{2}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} (-1)^{z \cdot y} |y\rangle$$
$$= \frac{1}{\sqrt{2^{n-1}}} \sum_{y \in \{0,1\}^n} (-1)^{z \cdot y} |y\rangle$$

- So when $a \neq 0^n$, the result will always be a string $y \in \{0,1\}^n : a \cdot y = 0$
- The amplitude associated with each value y is $\pm\sqrt{2^{1-n}}$, giving the probability:

$$\left|\frac{1}{\sqrt{2^{n-1}}}\right|^2 = \left|\frac{-1}{\sqrt{2^{n-1}}}\right|^2 = \frac{1}{2^{n-1}} \quad (8)$$

  of observing any of the strings y such that $a \cdot y = 0$
- A uniform distribution over the 2^{n-1} strings that satisfy $a \cdot y = 0$.

Post-processing: solving a system of linear equations

- If Simon's algorithm is executed $n-1$ times, $n-1$ strings y_1, y_2, ..., $y_{n-1} \in \{0,1\}^n$ can be observed, which form a system of $n-1$ linear equations in n unknowns of the form:

$$\begin{aligned} y_1 \cdot a &= y_{11}a_1 + y_{12}a_2 + \ldots + y_{1n}a_n = 0 \\ y_2 \cdot a &= y_{11}a_1 + y_{22}a_2 + \ldots + y_{2n}a_n = 0 \\ &\vdots \\ y_{n-1} \cdot a &= y_{(n-1)1}a_1 + y_{(n-1)2}a_2 + \ldots + y_{(n-1)n}a_n = 0 \end{aligned}$$

- To find a from here is just a matter of solving for the n unknowns, each a bit in a, in order to determine a as a whole
- Of course, this requires a system of $n-1$ linearly independent equations.

How to get a solvable system?

- The probability of observing the first string y_0 is 2^{1-n}
- After another iteration of Simon's algorithm, the probability of observing another distinct bit string would be $1 - 2^{1-n}$
- The probability of observing $n-1$ distinct values of y in a row, and so a lower bound on the probability of obtaining $n-1$ linearly independent equations, is:

$$\prod_{n=1}^{\infty} \left[1 - \frac{1}{2^n}\right] \approx .2887881 > \frac{1}{4}$$

- A linearly independent system of $n-1$ equations, and from there the value of a, can be obtained by repeating Simon's algorithm no more than $4n$ times
- Simon's algorithm requires only $O(n)$ queries to f in order to determine a, while classical algorithms require exponential time

Outline

Grover's Algorithm

- Quantum search
- How it works
- A worked example

Simon's algorithm

- Period-finding
- How it works
- An example

A 3-qubit example

- Now a worked example with $n = 3$, $a = 110$, and $f(x)$ defined by the following table:

x	$f(x)$
000	101
001	010
010	000
011	110
100	000
101	110
110	101
111	010

-

-

Image Credits

- Quantum dots:
  http://spectrum.ieee.org/nanoclast/semiconductors/nanotechnology/the-road-to-a-quantum-computer-begins-with-a-quantum-dot
- Lov Grover: http://www.bell-labs.com/user/lkgrover/
- Jacques Hadamard
  http://www.math.uconn.edu/MathLinks/mathematicians_gallery.php?