Quantum computation and Shor's factoring algorithm

Ronald de Wolf

Centrum voor Wiskunde en Informatica

Amsterdam

Quantum mechanics

Quantum mechanics

- Developed in the first half of 20th century

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

  superposition

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

  superposition, interference

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

  superposition, interference, entanglement

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

  superposition, interference, entanglement

- What about a quantum computer?

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

  superposition, interference, entanglement

- What about a quantum computer?

  1. can we build it?

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

  superposition, interference, entanglement

- What about a quantum computer?

  1. can we build it?
  2. what can it do?

Quantum mechanics

- Developed in the first half of 20th century
- One of our best physical theories
- Many “weird” effects:

  superposition, interference, entanglement

- What about a quantum computer?

  1. can we build it?
  2. what can it do?

- We focus on second question: quantum algorithms

Overview

Overview

- What are classical algorithms?

Overview

- What are classical algorithms?
- What are quantum algorithms?

Overview

- What are classical algorithms?
- What are quantum algorithms?
- Shor's algorithm for factoring integers (1994)

Overview

- What are classical algorithms?
- What are quantum algorithms?
- Shor's algorithm for factoring integers (1994)
- Other quantum algorithms

Classical algorithms

Classical algorithms

- Operate on bits

Classical algorithms

- Operate on bits
- Two main models:

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines and Boolean circuits

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines and Boolean circuits
- Circuits easier to generalize to quantum

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines and Boolean circuits
- Circuits easier to generalize to quantum
- Directed acyclic graph of AND, OR, NOT gates

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines and Boolean circuits
- Circuits easier to generalize to quantum
- Directed acyclic graph of AND, OR, NOT gates
- Starting nodes: n input bits, additional workspace

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines and Boolean circuits
- Circuits easier to generalize to quantum
- Directed acyclic graph of AND, OR, NOT gates
- Starting nodes: n input bits, additional workspace
- Compute $f : \{0,1\}^n \to \{0,1\}$ by evaluating all gates

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines and Boolean circuits
- Circuits easier to generalize to quantum
- Directed acyclic graph of AND, OR, NOT gates
- Starting nodes: n input bits, additional workspace
- Compute $f : \{0,1\}^n \to \{0,1\}$ by evaluating all gates
- Family $\{C_n\}$ of circuits (one for each input length n) can compute some problem on varying input length

Classical algorithms

- Operate on bits
- Two main models:
  Turing machines and Boolean circuits
- Circuits easier to generalize to quantum
- Directed acyclic graph of AND, OR, NOT gates
- Starting nodes: n input bits, additional workspace
- Compute $f : \{0,1\}^n \rightarrow \{0,1\}$ by evaluating all gates
- Family $\{C_n\}$ of circuits (one for each input length n) can compute some problem on varying input length
- Efficient if $|C_n| \leq poly(n)$

From classical to quantum

From classical to quantum

- bits $\longrightarrow$ qubits

From classical to quantum

- bits $\longrightarrow$ qubits
- AND/OR/NOT gates $\longrightarrow$ unitary quantum gates

From classical to quantum

- bits $\longrightarrow$ qubits
- AND/OR/NOT gates $\longrightarrow$ unitary quantum gates
- classical circuit $\longrightarrow$ quantum circuit

From classical to quantum

- bits $\longrightarrow$ qubits
- AND/OR/NOT gates $\longrightarrow$ unitary quantum gates
- classical circuit $\longrightarrow$ quantum circuit
- reading the output bit $\longrightarrow$ measuring final state

Quantum states and dynamics

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle$

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$
- n-qubit state: $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$
- n-qubit state: $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle = \begin{pmatrix} \alpha_{0\ldots0} \\ \vdots \\ \alpha_{1\ldots1} \end{pmatrix}$

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$
- n-qubit state: $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle = \begin{pmatrix} \alpha_{0\ldots0} \\ \vdots \\ \alpha_{1\ldots1} \end{pmatrix}$
- *Informally*: we are in all 2^n basis states simultaneously

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$
- n-qubit state: $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle = \begin{pmatrix} \alpha_{0\ldots0} \\ \vdots \\ \alpha_{1\ldots1} \end{pmatrix}$
- *Informally*: we are in all 2^n basis states simultaneously
- *Formally*: $|\phi\rangle$ is a vector in 2^n-dimensional Hilbert space

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$
- n-qubit state: $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle = \begin{pmatrix} \alpha_{0\ldots0} \\ \vdots \\ \alpha_{1\ldots1} \end{pmatrix}$
- *Informally*: we are in all 2^n basis states simultaneously
- *Formally*: $|\phi\rangle$ is a vector in 2^n-dimensional Hilbert space
- Two kinds of quantum operations on $|\phi\rangle$:

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$
- n-qubit state: $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle = \begin{pmatrix} \alpha_{0\ldots0} \\ \vdots \\ \alpha_{1\ldots1} \end{pmatrix}$
- *Informally*: we are in all 2^n basis states simultaneously
- *Formally*: $|\phi\rangle$ is a vector in 2^n-dimensional Hilbert space

- Two kinds of quantum operations on $|\phi\rangle$:
  1. Unitary transform of the amplitude-vector

Quantum states and dynamics

- Qubit: superposition $\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$
- n-qubit state: $|\phi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \begin{pmatrix} \alpha_{0\ldots0} \\ \vdots \\ \alpha_{1\ldots1} \end{pmatrix}$
- *Informally*: we are in all 2^n basis states simultaneously
- *Formally*: $|\phi\rangle$ is a vector in 2^n-dimensional Hilbert space

- Two kinds of quantum operations on $|\phi\rangle$:
  1. Unitary transform of the amplitude-vector
  2. Measurement gives $|x\rangle$ with probability $|\alpha_x|^2$

Quantum gates

Quantum gates

- Unitary transformations on 1 or 2 qubits

Quantum gates

- Unitary transformations on 1 or 2 qubits
- G on the first qubit: unitary $G \otimes I \otimes \cdots \otimes I$ on n qubits

Quantum gates

- Unitary transformations on 1 or 2 qubits
- G on the first qubit: unitary $G \otimes I \otimes \cdots \otimes I$ on n qubits
- 1-qubit Hadamard gate: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

Quantum gates

- Unitary transformations on 1 or 2 qubits
- G on the first qubit: unitary $G \otimes I \otimes \cdots \otimes I$ on n qubits
- 1-qubit Hadamard gate: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- 1-qubit $\pi/4$-gate: $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$

Quantum gates

- Unitary transformations on 1 or 2 qubits
- G on the first qubit: unitary $G \otimes I \otimes \cdots \otimes I$ on n qubits
- 1-qubit Hadamard gate: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
- 1-qubit $\pi/4$-gate: $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$
- 2-qubit controlled-NOT: $C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

Quantum circuits

Quantum circuits

- Circuit of quantum gates is unitary

Quantum circuits

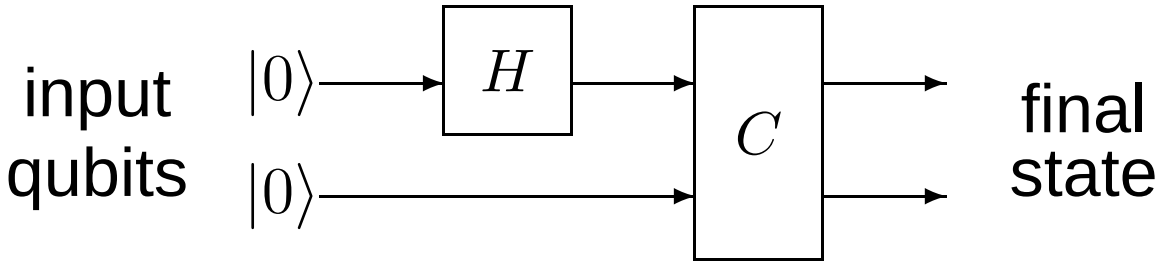
- Circuit of quantum gates is unitary
- Transforms (classical) input state to final state

Quantum circuits

- Circuit of quantum gates is unitary
- Transforms (classical) input state to final state
- Measure specific qubit of final state to obtain output

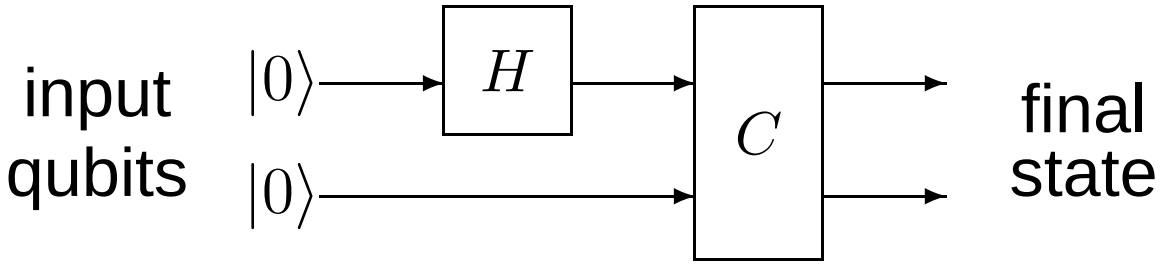
Quantum circuits

- Circuit of quantum gates is unitary
- Transforms (classical) input state to final state
- Measure specific qubit of final state to obtain output



Quantum circuits

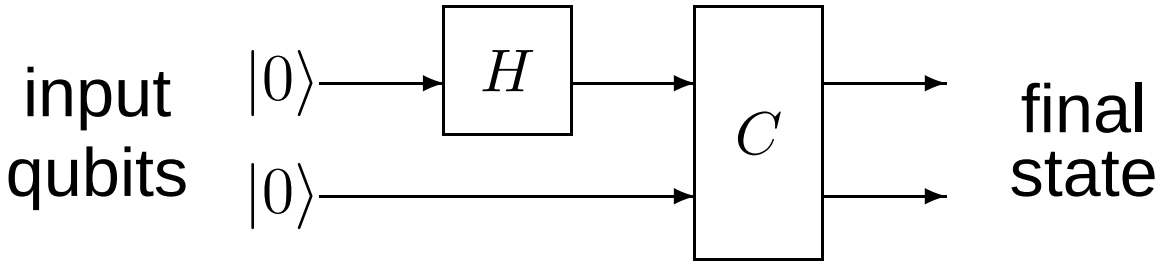
- Circuit of quantum gates is unitary
- Transforms (classical) input state to final state
- Measure specific qubit of final state to obtain output



- Final state: $\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$

Quantum circuits

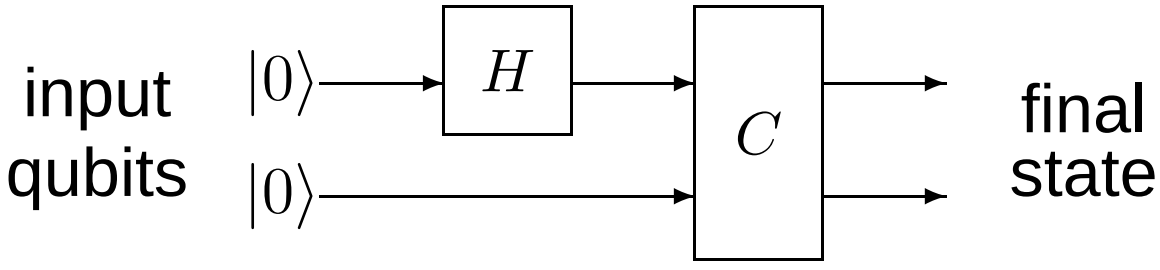
- Circuit of quantum gates is unitary
- Transforms (classical) input state to final state
- Measure specific qubit of final state to obtain output



- Final state: $\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$ (EPR-state)

Quantum circuits

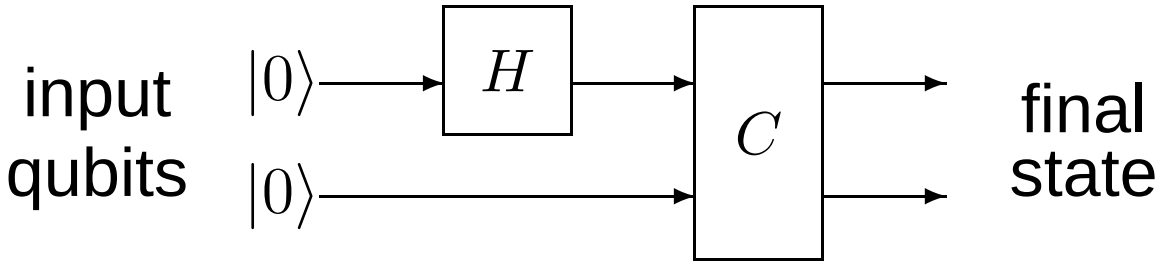
- Circuit of quantum gates is unitary
- Transforms (classical) input state to final state
- Measure specific qubit of final state to obtain output



- Final state: $\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$ (EPR-state)
- H, T, C gates can approximate any n-qubit unitary

Quantum circuits

- Circuit of quantum gates is unitary
- Transforms (classical) input state to final state
- Measure specific qubit of final state to obtain output



- Final state: $\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$ (EPR-state)
- H, T, C gates can approximate any n-qubit unitary
- Efficient quantum computation: polynomial family

Quantum parallelism

Quantum parallelism

- Suppose classical algorithm computes
  $f : \{0,1\}^n \rightarrow \{0,1\}^m$

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

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

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

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

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

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

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

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

- This contains all 2^n function values!

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

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

- This contains all 2^n function values!
- But observing gives only one random $|x\rangle|f(x)\rangle$

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

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

- This contains all 2^n function values!
- But observing gives only one random $|x\rangle|f(x)\rangle$
- All other information will be lost

Quantum parallelism

- Suppose classical algorithm computes $f : \{0,1\}^n \to \{0,1\}^m$
- Then quantum circuit $U : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$ can compute f on all inputs simultaneously!

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

- This contains all 2^n function values!
- But observing gives only one random $|x\rangle|f(x)\rangle$
- All other information will be lost
- More tricks needed for successful quantum computation

Factoring

Factoring

- Given $N = p \cdot q$, compute p and q

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem
- Fundamental computational problem on $\log N$ bits

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem
- Fundamental computational problem on $\log N$ bits
- Best known classical algorithms use time $2^{(\log N)^{\alpha}}$, where $\alpha = 1/2$ or $1/3$

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem
- Fundamental computational problem on $\log N$ bits
- Best known classical algorithms use time $2^{(\log N)^\alpha}$, where $\alpha = 1/2$ or $1/3$
- Its assumed computational hardness is basis of public-key cryptography (RSA):

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem
- Fundamental computational problem on $\log N$ bits
- Best known classical algorithms use time $2^{(\log N)^\alpha}$, where $\alpha = 1/2$ or $1/3$
- Its assumed computational hardness is basis of public-key cryptography (RSA):

  N is public key for encrypting a message

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem
- Fundamental computational problem on $\log N$ bits
- Best known classical algorithms use time $2^{(\log N)^\alpha}$, where $\alpha = 1/2$ or $1/3$
- Its assumed computational hardness is basis of public-key cryptography (RSA):

  N is public key for encrypting a message,
  p, q are private key for decrypting

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem
- Fundamental computational problem on $\log N$ bits
- Best known classical algorithms use time $2^{(\log N)^\alpha}$, where $\alpha = 1/2$ or $1/3$
- Its assumed computational hardness is basis of public-key cryptography (RSA):

  N is public key for encrypting a message,
  p, q are private key for decrypting

- A quantum computer can break this

Factoring

- Given $N = p \cdot q$, compute p and q
- Fundamental mathematical problem
- Fundamental computational problem on $\log N$ bits
- Best known classical algorithms use time $2^{(\log N)^\alpha}$, where $\alpha = 1/2$ or $1/3$
- Its assumed computational hardness is basis of public-key cryptography (RSA):

  N is public key for encrypting a message,
  p, q are private key for decrypting

- A quantum computer can break this,
  using Shor's efficient quantum factoring algorithm!

Reduction to period-finding

Reduction to period-finding

- Pick a random integer $x \in (0, N)$

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)
- For at least $1/4$ of the x's:

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)
- For at least $1/4$ of the x's:
  r is even and $x^{r/2} \pm 1 \not\equiv 0$ mod N
- Then:

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)
- For at least $1/4$ of the x's:
  r is even and $x^{r/2} \pm 1 \not\equiv 0$ mod N
- Then:
  $$x^r = (x^{r/2})^2 \equiv 1 \text{ mod } N$$

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)
- For at least $1/4$ of the x's:
  r is even and $x^{r/2} \pm 1 \not\equiv 0$ mod N
- Then:
  $$x^r = (x^{r/2})^2 \equiv 1 \text{ mod } N \iff$$
  $(x^{r/2} + 1)(x^{r/2} - 1) \equiv 0$ mod N

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)
- For at least $1/4$ of the x's:
  r is even and $x^{r/2} \pm 1 \not\equiv 0$ mod N
- Then:

$$x^r = (x^{r/2})^2 \equiv 1 \text{ mod } N \iff$$
$$(x^{r/2} + 1)(x^{r/2} - 1) \equiv 0 \text{ mod } N \iff$$
$$(x^{r/2} + 1)(x^{r/2} - 1) = kN \text{ for some } k$$

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)
- For at least $1/4$ of the x's:
  r is even and $x^{r/2} \pm 1 \not\equiv 0$ mod N
- Then:

$$x^r = (x^{r/2})^2 \equiv 1 \text{ mod } N \iff$$
$$(x^{r/2} + 1)(x^{r/2} - 1) \equiv 0 \text{ mod } N \iff$$
$$(x^{r/2} + 1)(x^{r/2} - 1) = kN \text{ for some } k$$

- $x^{r/2} \pm 1$ shares a factor with N

Reduction to period-finding

- Pick a random integer $x \in (0, N)$
- The sequence $x^0, x^1, x^2, x^3, \ldots$ mod N cycles
- Unknown period r (least $r > 0$ s.t. $x^r \equiv 1$ mod N)
- For at least $1/4$ of the x's:
  r is even and $x^{r/2} \pm 1 \not\equiv 0$ mod N
- Then:
  $$x^r = (x^{r/2})^2 \equiv 1 \text{ mod } N \iff$$
  $$(x^{r/2} + 1)(x^{r/2} - 1) \equiv 0 \text{ mod } N \iff$$
  $$(x^{r/2} + 1)(x^{r/2} - 1) = kN \text{ for some } k$$
- $x^{r/2} \pm 1$ shares a factor with N
- This factor of N can be extracted using gcd-algorithm

Overview of Shor's algorithm

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N
- Shor's quantum algorithm for period-finding

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N
- Shor's quantum algorithm for period-finding

  (2 registers, initially $|\underbrace{0\ldots0}_{\log q}\rangle|\underbrace{0\ldots0}_{\log N}\rangle$)

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N
- Shor's quantum algorithm for period-finding

  (2 registers, initially $|\underbrace{0\ldots0}_{\log q}\rangle|\underbrace{0\ldots0}_{\log N}\rangle$)

  1. Apply quantum Fourier transform to 1st register

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N
- Shor's quantum algorithm for period-finding

  (2 registers, initially $|\underbrace{0\ldots0}_{\log q}\rangle|\underbrace{0\ldots0}_{\log N}\rangle$)

  1. Apply quantum Fourier transform to 1st register
  2. Compute $|a\rangle|0\rangle \mapsto |a\rangle|x^a \text{ mod } N\rangle$ in quantum parallel

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N
- Shor's quantum algorithm for period-finding

  (2 registers, initially $|\underbrace{0\ldots0}_{\log q}\rangle|\underbrace{0\ldots0}_{\log N}\rangle$)

  1. Apply quantum Fourier transform to 1st register
  2. Compute $|a\rangle|0\rangle \mapsto |a\rangle|x^a \text{ mod } N\rangle$ in quantum parallel
  3. Measure 2nd register

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N
- Shor's quantum algorithm for period-finding

  (2 registers, initially $|\underbrace{0\ldots0}_{\log q}\rangle|\underbrace{0\ldots0}_{\log N}\rangle$)

1. Apply quantum Fourier transform to 1st register
2. Compute $|a\rangle|0\rangle \mapsto |a\rangle|x^a \text{ mod } N\rangle$ in quantum parallel
3. Measure 2nd register
4. Apply quantum Fourier transform to 1st register

Overview of Shor's algorithm

- Given x and N, we want to find the period r of x, mod N
- Shor's quantum algorithm for period-finding

  (2 registers, initially $|\underbrace{0\ldots0}_{\log q}\rangle|\underbrace{0\ldots0}_{\log N}\rangle$)

1. Apply quantum Fourier transform to 1st register
2. Compute $|a\rangle|0\rangle \mapsto |a\rangle|x^a \text{ mod } N\rangle$ in quantum parallel
3. Measure 2nd register
4. Apply quantum Fourier transform to 1st register
5. Measure 1st register and computer r

Quantum Fourier transform

Quantum Fourier transform

- Fourier basis for space with dimension q:

Quantum Fourier transform

- Fourier basis for space with dimension q:

$$|\chi_a\rangle = \frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} e^{\frac{2\pi i a b}{q}} |b\rangle$$

Quantum Fourier transform

- Fourier basis for space with dimension q:

$$|\chi_a\rangle = \frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} e^{\frac{2\pi i a b}{q}} |b\rangle$$

- Quantum Fourier Transform: $|a\rangle \rightarrow |\chi_a\rangle$

Quantum Fourier transform

- Fourier basis for space with dimension q:

$$|\chi_a\rangle = \frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} e^{\frac{2\pi i a b}{q}} |b\rangle$$

- Quantum Fourier Transform: $|a\rangle \rightarrow |\chi_a\rangle$
- This is just the Fourier transform over cyclic group C_q

Quantum Fourier transform

- Fourier basis for space with dimension q:

$$|\chi_a\rangle = \frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} e^{\frac{2\pi i a b}{q}} |b\rangle$$

- Quantum Fourier Transform: $|a\rangle \rightarrow |\chi_a\rangle$
- This is just the Fourier transform over cyclic group C_q
- If $q = 2^k$, then we can implement this with a quantum circuit of only $O(k \log k)$ gates

Quantum Fourier transform

- Fourier basis for space with dimension q:

$$|\chi_a\rangle = \frac{1}{\sqrt{q}} \sum_{b=0}^{q-1} e^{\frac{2\pi i a b}{q}} |b\rangle$$

- Quantum Fourier Transform: $|a\rangle \rightarrow |\chi_a\rangle$
- This is just the Fourier transform over cyclic group C_q
- If $q = 2^k$, then we can implement this with a quantum circuit of only $O(k \log k)$ gates
- For our q, we choose the power of 2 in $(N^2, 2N^2]$

Easy case: $r \big| q$

Easy case: $r \big| q$

1. Apply QFT to 1st register of $|0 \ldots 0\rangle |0 \ldots 0\rangle$:

Easy case: $r \big| q$

1. Apply QFT to 1st register of $|0 \ldots 0\rangle |0 \ldots 0\rangle$:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |0\rangle$$

Easy case: $r \big| q$

1. Apply QFT to 1st register of $|0 \ldots 0\rangle|0 \ldots 0\rangle$:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0\rangle$$

2. Compute x^a mod N in parallel (Schönhage-Strassen)

Easy case: $r \big| q$

1. Apply QFT to 1st register of $|0 \ldots 0\rangle|0 \ldots 0\rangle$:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0\rangle$$

2. Compute x^a mod N in parallel (Schönhage-Strassen)

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|x^a \text{ mod } N\rangle$$

Easy case: $r|q$

1. Apply QFT to 1st register of $|0\ldots0\rangle|0\ldots0\rangle$:

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|0\rangle$$

2. Compute x^a mod N in parallel (Schönhage-Strassen)

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|x^a \text{ mod } N\rangle$$

3. Observing 2nd register gives $|x^s$ mod $N\rangle$ (random s),

Easy case: $r|q$

1. Apply QFT to 1st register of $|0\ldots0\rangle|0\ldots0\rangle$:

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|0\rangle$$

2. Compute x^a mod N in parallel (Schönhage-Strassen)

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|x^a \text{ mod } N\rangle$$

3. Observing 2nd register gives $|x^s$ mod $N\rangle$ (random s),
   1st register collapses to superposition of

$$|s\rangle, |r+s\rangle, |2r+s\rangle, \ldots, |q-r+s\rangle$$

Easy case: $r \big| q$ (continued)

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

$$\sum_{j=0}^{q/r-1} \sum_{b=0}^{q-1} e^{2\pi i \frac{(jr+s)b}{q}} |b\rangle$$

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

$$\sum_{j=0}^{q/r-1} \sum_{b=0}^{q-1} e^{2\pi i \frac{(jr+s)b}{q}} |b\rangle = \sum_{b=0}^{q-1} e^{2\pi i \frac{sb}{q}} \underbrace{\left(\sum_{j=0}^{q/r-1} e^{2\pi i \frac{jrb}{q}} \right)}_{\text{geometric sum}} |b\rangle$$

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

$$\sum_{j=0}^{q/r-1} \sum_{b=0}^{q-1} e^{2\pi i \frac{(jr+s)b}{q}} |b\rangle = \sum_{b=0}^{q-1} e^{2\pi i \frac{sb}{q}} \underbrace{\left(\sum_{j=0}^{q/r-1} e^{2\pi i \frac{jrb}{q}} \right)}_{\text{geometric sum}} |b\rangle$$

Sum

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

$$\sum_{j=0}^{q/r-1} \sum_{b=0}^{q-1} e^{2\pi i \frac{(jr+s)b}{q}} |b\rangle = \sum_{b=0}^{q-1} e^{2\pi i \frac{sb}{q}} \underbrace{\left(\sum_{j=0}^{q/r-1} e^{2\pi i \frac{jrb}{q}} \right)}_{\text{geometric sum}} |b\rangle$$

Sum $\neq 0$

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

$$\sum_{j=0}^{q/r-1} \sum_{b=0}^{q-1} e^{2\pi i \frac{(jr+s)b}{q}} |b\rangle = \sum_{b=0}^{q-1} e^{2\pi i \frac{sb}{q}} \underbrace{\left(\sum_{j=0}^{q/r-1} e^{2\pi i \frac{jrb}{q}} \right)}_{\text{geometric sum}} |b\rangle$$

Sum $\neq 0$ iff $e^{2\pi i \frac{rb}{q}} = 1$

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

$$\sum_{j=0}^{q/r-1} \sum_{b=0}^{q-1} e^{2\pi i \frac{(jr+s)b}{q}} |b\rangle = \sum_{b=0}^{q-1} e^{2\pi i \frac{sb}{q}} \underbrace{\left(\sum_{j=0}^{q/r-1} e^{2\pi i \frac{jrb}{q}} \right)}_{\text{geometric sum}} |b\rangle$$

Sum $\neq 0$ iff $e^{2\pi i \frac{rb}{q}} = 1$ iff $\frac{rb}{q}$ is an integer

Easy case: $r|q$ (continued)

Recall: 1st register is in superposition $\sum_{j=0}^{q/r-1} |jr+s\rangle$

4. Apply QFT once more:

$$\sum_{j=0}^{q/r-1}\sum_{b=0}^{q-1} e^{2\pi i\frac{(jr+s)b}{q}}|b\rangle = \sum_{b=0}^{q-1} e^{2\pi i\frac{sb}{q}} \underbrace{\left(\sum_{j=0}^{q/r-1} e^{2\pi i\frac{jrb}{q}}\right)}_{\text{geometric sum}} |b\rangle$$

Sum $\neq 0$ iff $e^{2\pi i\frac{rb}{q}} = 1$ iff $\frac{rb}{q}$ is an integer

Only the b that are multiples of $\frac{q}{r}$ have non-zero amplitude!

Easy case: $r|q$ (continued)

Easy case: $r|q$ (continued)

5. Observe 1st register: random multiple $b = c\frac{q}{r}$ ($c \in [0, r)$)

Easy case: $r|q$ (continued)

5. Observe 1st register: random multiple $b = c\frac{q}{r}$ ($c \in [0, r)$):

$$\frac{b}{q} = \frac{c}{r}$$

Easy case: $r|q$ (continued)

5. Observe 1st register: random multiple $b = c\frac{q}{r}$ ($c \in [0, r)$):

$$\frac{b}{q} = \frac{c}{r}$$

- b and q are known

Easy case: $r|q$ (continued)

5. Observe 1st register: random multiple $b = c\frac{q}{r}$ ($c \in [0, r)$):

$$\frac{b}{q} = \frac{c}{r}$$

- b and q are known; c and r are unknown

Easy case: $r|q$ (continued)

5. Observe 1st register: random multiple $b = c\frac{q}{r}$ ($c \in [0, r)$):

$$\frac{b}{q} = \frac{c}{r}$$

- b and q are known; c and r are unknown
- c and r are coprime with probability $\Omega(1/\log\log r)$

Easy case: $r|q$ (continued)

5. Observe 1st register: random multiple $b = c\frac{q}{r}$ ($c \in [0, r)$):

$$\frac{b}{q} = \frac{c}{r}$$

- b and q are known; c and r are unknown
- c and r are coprime with probability $\Omega(1/\log\log r)$
- Then: we know r by writing $\frac{b}{q}$ in lowest terms

Easy case: $r|q$ (continued)

5. Observe 1st register: random multiple $b = c\frac{q}{r}$ ($c \in [0, r)$):

$$\frac{b}{q} = \frac{c}{r}$$

- b and q are known; c and r are unknown
- c and r are coprime with probability $\Omega(1/\log\log r)$
- Then: we know r by writing $\frac{b}{q}$ in lowest terms
- Since we can find r, we can factor!

Other quantum algorithms

Other quantum algorithms

- Generalizations of Shor's algorithm:

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem
  2. Pell's equation (Hallgren 2002)

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem
  2. Pell's equation (Hallgren 2002)
- Quantum search algorithm (Grover 1996):
  Search N-element space in $\sqrt{N}$ steps.

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem
  2. Pell's equation (Hallgren 2002)
- Quantum search algorithm (Grover 1996):
  Search N-element space in $\sqrt{N}$ steps.
  Used for: minimum-finding, shortest paths, MST,. . .

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem
  2. Pell's equation (Hallgren 2002)
- Quantum search algorithm (Grover 1996):
  Search N-element space in $\sqrt{N}$ steps.
  Used for: minimum-finding, shortest paths, MST,. . .
- Quantum random walk algorithms:

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem
  2. Pell's equation (Hallgren 2002)
- Quantum search algorithm (Grover 1996):
  Search N-element space in $\sqrt{N}$ steps.
  Used for: minimum-finding, shortest paths, MST,...
- Quantum random walk algorithms:
  1. determine if $a_1, \ldots, a_N$ are all distinct in $N^{2/3}$ steps (Ambainis 2003)

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem
  2. Pell's equation (Hallgren 2002)
- Quantum search algorithm (Grover 1996):
  Search N-element space in $\sqrt{N}$ steps.
  Used for: minimum-finding, shortest paths, MST,...
- Quantum random walk algorithms:
  1. determine if $a_1, \ldots, a_N$ are all distinct in $N^{2/3}$ steps (Ambainis 2003)
  2. evaluate a gametree with N positions in $\sqrt{N}$ steps (Farhi, Goldstone, Gutmann 2007)

Other quantum algorithms

- Generalizations of Shor's algorithm:
  1. Hidden subgroup problem
  2. Pell's equation (Hallgren 2002)
- Quantum search algorithm (Grover 1996):
  Search N-element space in $\sqrt{N}$ steps.
  Used for: minimum-finding, shortest paths, MST,...
- Quantum random walk algorithms:
  1. determine if $a_1, \ldots, a_N$ are all distinct in $N^{2/3}$ steps (Ambainis 2003)
  2. evaluate a gametree with N positions in $\sqrt{N}$ steps (Farhi, Goldstone, Gutmann 2007)
- Jones polynomial (Aharonov et al.)

Summary

Summary

- We introduced the circuit model for quantum algorithms

Summary

- We introduced the circuit model for quantum algorithms
- Explained Shor's factoring algorithm

Summary

- We introduced the circuit model for quantum algorithms
- Explained Shor's factoring algorithm
- Mentioned a few others

Summary

- We introduced the circuit model for quantum algorithms
- Explained Shor's factoring algorithm
- Mentioned a few others
- There should be many more. . .