Quantum Computation

Lecture 27

And that's all we got time for!

State

State

- State of a classical computer labeled by (say) bit strings

State

- State of a classical computer labeled by (say) bit strings
  - e.g. 2-bit states: 00, 01, 10 and 11

State

- State of a classical computer labeled by (say) bit strings
  - e.g. 2-bit states: 00, 01, 10 and 11
- Probabilistic computation: state is a probability distribution over the basis states

State

- State of a classical computer labeled by (say) bit strings
  - e.g. 2-bit states: 00, 01, 10 and 11
- Probabilistic computation: state is a probability distribution over the basis states
  - $p = (p_{00},p_{01},p_{10},p_{11})$ s.t. p_{ij} non-negative and $||p||_1 = 1$

State

- State of a classical computer labeled by (say) bit strings
  - e.g. 2-bit states: 00, 01, 10 and 11
- Probabilistic computation: state is a probability distribution over the basis states
  - $p = (p_{00}, p_{01}, p_{10}, p_{11})$ s.t. p_{ij} non-negative and $\|p\|_1 = 1$
- Quantum computation/Quantum mechanics: state is a real (or even complex) vector

State

- State of a classical computer labeled by (say) bit strings
  - e.g. 2-bit states: 00, 01, 10 and 11
- Probabilistic computation: state is a probability distribution over the basis states
  - $p = (p_{00}, p_{01}, p_{10}, p_{11})$ s.t. p_{ij} non-negative and $||p||_1 = 1$
- Quantum computation/Quantum mechanics: state is a real (or even complex) vector
  - $q = (q_{00}, q_{01}, q_{10}, q_{11})$ s.t. $||q||_2 = 1$

State

- State of a classical computer labeled by (say) bit strings
  - e.g. 2-bit states: 00, 01, 10 and 11
- Probabilistic computation: state is a probability distribution over the basis states
  - $p = (p_{00},p_{01},p_{10},p_{11})$ s.t. p_{ij} non-negative and $||p||_1 = 1$
- Quantum computation/Quantum mechanics: state is a real (or even complex) vector
  - $q = (q_{00},q_{01},q_{10},q_{11})$ s.t. $||q||_2 = 1$
  - q_s is the “amplitude” of basis state s

Qubits

Qubits

- State of a quantum system is stored as qubits

Qubits

- State of a quantum system is stored as qubits
  - Physically, some property (spin, polarization) of a particle (electron, photon) that takes two discrete values

Qubits

- State of a quantum system is stored as qubits
  - Physically, some property (spin, polarization) of a particle (electron, photon) that takes two discrete values
- State of a single qubit: a 2-dimensional vector of unit L_2 norm

Qubits

- State of a quantum system is stored as qubits
  - Physically, some property (spin, polarization) of a particle (electron, photon) that takes two discrete values
- State of a single qubit: a 2-dimensional vector of unit L_2 norm
  - Joint state of independent qubits: tensor product of their individual states (like classical probability)

Qubits

- State of a quantum system is stored as qubits
  - Physically, some property (spin, polarization) of a particle (electron, photon) that takes two discrete values
- State of a single qubit: a 2-dimensional vector of unit L_2 norm
  - Joint state of independent qubits: tensor product of their individual states (like classical probability)
- An m qubit system has 2^m basis states. Its quantum state can be any valid amplitude vector (2^m dimensional complex vector, with unit L_2 norm), not always separable into independent qubits

Qubits

- State of a quantum system is stored as qubits
  - Physically, some property (spin, polarization) of a particle (electron, photon) that takes two discrete values
- State of a single qubit: a 2-dimensional vector of unit L_2 norm
  - Joint state of independent qubits: tensor product of their individual states (like classical probability)
- An m qubit system has 2^m basis states. Its quantum state can be any valid amplitude vector (2^m dimensional complex vector, with unit L_2 norm), not always separable into independent qubits
  - e.g. $\sqrt{½}$ [1 0 0 -1]. Also written as $\sqrt{½}\,|00\rangle - \sqrt{½}\,|11\rangle$

Qubits

- State of a quantum system is stored as qubits
  - Physically, some property (spin, polarization) of a particle (electron, photon) that takes two discrete values
- State of a single qubit: a 2-dimensional vector of unit L_2 norm
  - Joint state of independent qubits: tensor product of their individual states (like classical probability)
- An m qubit system has 2^m basis states. Its quantum state can be any valid amplitude vector (2^m dimensional complex vector, with unit L_2 norm), not always separable into independent qubits
  - e.g. $\sqrt{½}\begin{array}{cccccc} & |00\rangle & |01\rangle & |10\rangle & |11\rangle & \\ [& 1 & 0 & 0 & -1 &] \end{array}$. Also written as $\sqrt{½}\,|00\rangle - \sqrt{½}\,|11\rangle$

Qubits

- State of a quantum system is stored as qubits
  - Physically, some property (spin, polarization) of a particle (electron, photon) that takes two discrete values
- State of a single qubit: a 2-dimensional vector of unit L_2 norm
  - Joint state of independent qubits: tensor product of their individual states (like classical probability)
- An m qubit system has 2^m basis states. Its quantum state can be any valid amplitude vector (2^m dimensional complex vector, with unit L_2 norm), not always separable into independent qubits
  - e.g. $\sqrt{\tfrac{1}{2}}\ \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ [\ 1 & 0 & 0 & -1\ ] \end{array}$. Also written as $\sqrt{\tfrac{1}{2}}\,|00\rangle - \sqrt{\tfrac{1}{2}}\,|11\rangle$
- (Also, state can be “mixed”: a probability distribution over amplitude vectors. Doesn’t change power of quantum computing)

Measuring a Quantum state

Measuring a Quantum state

- Measuring a state outputs one of the basis states, and the original state collapses to that basis state

Measuring a Quantum state

- Measuring a state outputs one of the basis states, and the original state collapses to that basis state
  - Probability of getting state |i⟩ is the square of its amplitude

Measuring a Quantum state

- Measuring a state outputs one of the basis states, and the original state collapses to that basis state
    - Probability of getting state |i⟩ is the square of its amplitude
    - Let's call the amplitude-square vector the measurement

Measuring a Quantum state

- Measuring a state outputs one of the basis states, and the original state collapses to that basis state
  - Probability of getting state |i⟩ is the square of its amplitude
  - Let’s call the amplitude-square vector the measurement
    - Measurement is a probability distribution over possible outcomes (namely the basis states)

Measuring a Quantum state

- Measuring a state outputs one of the basis states, and the original state collapses to that basis state
  - Probability of getting state |i⟩ is the square of its amplitude
  - Let's call the amplitude-square vector the measurement
    - Measurement is a probability distribution over possible outcomes (namely the basis states)
- Can do partial measurement - i.e., measurement on some qubits only - and continue computing. State collapses to be consistent with the measurement

Measuring a Quantum state

- Measuring a state outputs one of the basis states, and the original state collapses to that basis state
  - Probability of getting state |i> is the square of its amplitude
  - Let's call the amplitude-square vector the measurement
    - Measurement is a probability distribution over possible outcomes (namely the basis states)
- Can do partial measurement - i.e., measurement on some qubits only - and continue computing. State collapses to be consistent with the measurement
  - Can modify computation to defer all measurements to the end

Measuring a Quantum state

- Measuring a state outputs one of the basis states, and the original state collapses to that basis state
  - Probability of getting state $|i\rangle$ is the square of its amplitude
  - Let's call the amplitude-square vector the measurement
    - Measurement is a probability distribution over possible outcomes (namely the basis states)
- Can do partial measurement - i.e., measurement on some qubits only - and continue computing. State collapses to be consistent with the measurement
  - Can modify computation to defer all measurements to the end
- Can choose "non-standard" bases for measurement. But again, can do without it

Operations on state

Operations on state

- Unitary operations: linear transforms that preserve the L_2 norm

Operations on state

- Unitary operations: linear transforms that preserve the L_2 norm
  - Multiplication by a unitary matrix: i.e., $U^\dagger = U^{-1}$

Operations on state

- Unitary operations: linear transforms that preserve the L_2 norm
  - Multiplication by a unitary matrix: i.e., $U^\dagger = U^{-1}$

Conjugate transpose

Operations on state

- Unitary operations: linear transforms that preserve the L_2 norm
  - Multiplication by a unitary matrix: i.e., $U^\dagger = U^{-1}$
  - For quantum computing can restrict to real matrices

Conjugate transpose

Operations on state

- Unitary operations: linear transforms that preserve the L_2 norm
  - Multiplication by a unitary matrix: i.e., $U^\dagger = U^{-1}$
  - For quantum computing can restrict to real matrices
- Unitary matrices are invertible

Conjugate transpose

Operations on state

- Unitary operations: linear transforms that preserve the L_2 norm
  - Multiplication by a unitary matrix: i.e., $U^\dagger = U^{-1}$
  - For quantum computing can restrict to real matrices

Conjugate transpose

- Unitary matrices are invertible
  - Computation is reversible!

Operations on state

- Unitary operations: linear transforms that preserve the L_2 norm
  - Multiplication by a **unitary matrix**: i.e., $U^\dagger = U^{-1}$
  - For quantum computing can restrict to **real** matrices

Conjugate transpose

- Unitary matrices are invertible
  - Computation is reversible!
- e.g.: $\begin{matrix} \sqrt{\tfrac{1}{2}} & \sqrt{\tfrac{1}{2}} \\ \sqrt{\tfrac{1}{2}} & -\sqrt{\tfrac{1}{2}} \end{matrix}$ (on one qubit), $\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{matrix}$ (on 2 qubits)

Operations on state

Operations on state

- Hadamard transform (on a single qubit)

Operations on state

- Hadamard transform (on a single qubit)
  - Takes [1 0] to $\sqrt{½}$ [1 1], and [0 1] to $\sqrt{½}$ [1 -1]

Operations on state

- Hadamard transform (on a single qubit)
  - Takes [1 0] to $\sqrt{½}$ [1 1], and [0 1] to $\sqrt{½}$ [1 -1]
  - Measurement of result of applying this to a basis state is [½ ½] (i.e., can be used to toss a coin)

Operations on state

- Hadamard transform (on a single qubit)
  - Takes [1 0] to √½ [1 1], and [0 1] to √½ [1 -1]
  - Measurement of result of applying this to a basis state is [½ ½] (i.e., can be used to toss a coin)
- A quantum effect:

Operations on state

- Hadamard transform (on a single qubit)
    - Takes [1 0] to √½ [1 1], and [0 1] to √½ [1 -1]
    - Measurement of result of applying this to a basis state is [½ ½] (i.e., can be used to toss a coin)
- A quantum effect:
    - Had([1 0]) = [√½ √½]; Had([√½ √½]) = [1 0].

Operations on state

- Hadamard transform (on a single qubit)
  - Takes [1 0] to √½ [1 1], and [0 1] to √½ [1 -1]
  - Measurement of result of applying this to a basis state is [½ ½] (i.e., can be used to toss a coin)
- A quantum effect:
  - Had([1 0]) = [√½ √½]; Had([√½ √½]) = [1 0].
    - Amplitudes of |1⟩ destructively interfere!

Operations on state

- Hadamard transform (on a single qubit)
  - Takes [1 0] to √½ [1 1], and [0 1] to √½ [1 -1]
  - Measurement of result of applying this to a basis state is [½ ½] (i.e., can be used to toss a coin)
- A quantum effect:
  - Had([1 0]) = [√½ √½]; Had([√½ √½]) = [1 0].
    - Amplitudes of |1❯ destructively interfere!
    - Contrast with classical case: probabilities can only add

Quantum gates

Quantum gates

- A quantum gate: Unitary operation on a small number of (say three) qubits

Quantum gates

- A quantum gate: Unitary operation on a small number of (say three) qubits
  - Number of input qubits equals number of output qubits

Quantum gates

- A quantum gate: Unitary operation on a small number of (say three) qubits
  - Number of input qubits equals number of output qubits
  - There are infinitely many quantum gates

Quantum gates

- A quantum gate: Unitary operation on a small number of (say three) qubits
  - Number of input qubits equals number of output qubits
  - There are infinitely many quantum gates
- A universal set of gates: can be used to well approximate any gate

Quantum gates

- A quantum gate: Unitary operation on a small number of (say three) qubits
  - Number of input qubits equals number of output qubits
  - There are infinitely many quantum gates
- A universal set of gates: can be used to well approximate any gate
  - e.g. Hadamard gate and Toffoli gate (when restricted to real amplitudes)

Quantum gates

- A quantum gate: Unitary operation on a small number of (say three) qubits
  - Number of input qubits equals number of output qubits
  - There are infinitely many quantum gates
- A universal set of gates: can be used to well approximate any gate
  - e.g. Hadamard gate and Toffoli gate (when restricted to real amplitudes)
  - Toffoli gate has a classical analog (on 3 bits) that can be described as $T(a,b,c) = (a,b,c\oplus a\wedge b)$

Cleaning up the Garbage

Cleaning up the Garbage

- Since only reversible gates, need extra qubits (scratch space) as input and output

Cleaning up the Garbage

- Since only reversible gates, need extra qubits (scratch space) as input and output
- At the output, the values of this scratch qubits will depend on the input more than the desired output depends on it

Cleaning up the Garbage

- Since only reversible gates, need extra qubits (scratch space) as input and output
- At the output, the values of this scratch qubits will depend on the input more than the desired output depends on it
  - "Garbage"

Cleaning up the Garbage

- Since only reversible gates, need extra qubits (scratch space) as input and output
- At the output, the values of this scratch qubits will depend on the input more than the desired output depends on it
  - “Garbage”
- Can be a problem: e.g., two amplitudes will not cancel out because their garbage values are different

Cleaning up the Garbage

- Since only reversible gates, need extra qubits (scratch space) as input and output
- At the output, the values of this scratch qubits will depend on the input more than the desired output depends on it
  - "Garbage"
- Can be a problem: e.g., two amplitudes will not cancel out because their garbage values are different
- Solution: Ensure garbage qubits are returned to a standard state, by "uncomputing"

Cleaning up the Garbage

- Since only reversible gates, need extra qubits (scratch space) as input and output
- At the output, the values of this scratch qubits will depend on the input more than the desired output depends on it
  - "Garbage"
- Can be a problem: e.g., two amplitudes will not cancel out because their garbage values are different
- Solution: Ensure garbage qubits are returned to a standard state, by "uncomputing"
  - "Copy" the output to unused qubits, and run the reverse computation to return the rest to original state

Quantum Circuits and BQP

Quantum Circuits and BQP

- Quantum circuit: composed of quantum gates

Quantum Circuits and BQP

- Quantum circuit: composed of quantum gates
  - And a quantum measurement at the end

Quantum Circuits and BQP

- Quantum circuit: composed of quantum gates
  - And a quantum measurement at the end
- To decide a language measurement on a single qubit

Quantum Circuits and BQP

- Quantum circuit: composed of quantum gates
  - And a quantum measurement at the end
- To decide a language measurement on a single qubit
- We shall require a poly-time uniform circuit family

Quantum Circuits and BQP

- Quantum circuit: composed of quantum gates
    - And a quantum measurement at the end
- To decide a language measurement on a single qubit
- We shall require a poly-time uniform circuit family
    - It should be possible for a (classical/deterministic) TM to efficiently output the description of the quantum circuit for any given input length

Quantum Circuits and BQP

- Quantum circuit: composed of quantum gates
  - And a quantum measurement at the end
- To decide a language measurement on a single qubit
- We shall require a poly-time uniform circuit family
  - It should be possible for a (classical/deterministic) TM to efficiently output the description of the quantum circuit for any given input length
- BQP: Class of languages L for which there is a poly-sized (and poly-time uniform) quantum circuit family $\{C_n\}$ s.t. for all n, for all x, |x|=n,

Quantum Circuits and BQP

- Quantum circuit: composed of quantum gates
  - And a quantum measurement at the end
- To decide a language measurement on a single qubit
- We shall require a poly-time uniform circuit family
  - It should be possible for a (classical/deterministic) TM to efficiently output the description of the quantum circuit for any given input length
- BQP: Class of languages L for which there is a poly-sized (and poly-time uniform) quantum circuit family $\{C_n\}$ s.t. for all n, for all x, $|x|=n$,
  - $x \in L \Rightarrow C_n(|x0^m\rangle) = 1$ w.p. $> 2/3$; $x \notin L \Rightarrow C_n(|x0^m\rangle) = 1$ w.p. $< 1/3$

BQP

BQP

- BPP $\subseteq$ BQP: Classical gates and coin-flipping can be emulated by quantum gates

BQP

- BPP $\subseteq$ BQP: Classical gates and coin-flipping can be emulated by quantum gates
- Probability of a quantum circuit (with say Hadamard and Toffoli gates) accepting can be calculated classically, by brute force

BQP

- BPP $\subseteq$ BQP: Classical gates and coin-flipping can be emulated by quantum gates
- Probability of a quantum circuit (with say Hadamard and Toffoli gates) accepting can be calculated classically, by brute force
  - Multiply together all 2^nx2^n unitary matrices in EXP

BQP

- BPP ⊆ BQP: Classical gates and coin-flipping can be emulated by quantum gates
- Probability of a quantum circuit (with say Hadamard and Toffoli gates) accepting can be calculated classically, by brute force
  - Multiply together all 2^nx2^n unitary matrices in EXP
  - More carefully, since each gate involves only 3 qubits, in PSPACE

BQP

- BPP ⊆ BQP: Classical gates and coin-flipping can be emulated by quantum gates
- Probability of a quantum circuit (with say Hadamard and Toffoli gates) accepting can be calculated classically, by brute force
  - Multiply together all 2^nx2^n unitary matrices in EXP
  - More carefully, since each gate involves only 3 qubits, in PSPACE
  - In fact, can be done in PP. i.e., BQP ⊆ PP

BQP

- BPP ⊆ BQP: Classical gates and coin-flipping can be emulated by quantum gates
- Probability of a quantum circuit (with say Hadamard and Toffoli gates) accepting can be calculated classically, by brute force
  - Multiply together all 2^nx2^n unitary matrices in EXP
  - More carefully, since each gate involves only 3 qubits, in PSPACE
  - In fact, can be done in PP. i.e., BQP ⊆ PP
- How about BQP and NP?

Two Quantum Algorithms

Two Quantum Algorithms

- Grover's Search

Two Quantum Algorithms

- Grover's Search
  - Quadratic speedup for NP-complete problems (over the best known classical algorithms)

Two Quantum Algorithms

- Grover's Search
  - Quadratic speedup for NP-complete problems (over the best known classical algorithms)
  - Solve any NP problem with $O(2^{n/2})$ quantum gate operations

Two Quantum Algorithms

- Grover's Search
  - Quadratic speedup for NP-complete problems (over the best known classical algorithms)
  - Solve any NP problem with $O(2^{n/2})$ quantum gate operations
- Shor's Factoring

Two Quantum Algorithms

- Grover's Search
  - Quadratic speedup for NP-complete problems (over the best known classical algorithms)
  - Solve any NP problem with $O(2^{n/2})$ quantum gate operations
- Shor's Factoring
  - Polynomial sized quantum circuit for factoring

Two Quantum Algorithms

- Grover's Search
  - Quadratic speedup for NP-complete problems (over the best known classical algorithms)
  - Solve any NP problem with $O(2^{n/2})$ quantum gate operations
- Shor's Factoring
  - Polynomial sized quantum circuit for factoring
  - Exponential speedup over the best known classical algorithms

Grover's Search

Grover's Search

- Suppose f has a unique satisfying input z

Grover's Search

- Suppose f has a unique satisfying input z
  - Otherwise, modify f (by adding a hash "filter") so that with good probability it has a unique solution (if any)

Grover's Search

- Suppose f has a unique satisfying input z
  - Otherwise, modify f (by adding a hash "filter") so that with good probability it has a unique solution (if any)
- Plan: start with the uniform superposition on n-qubits (i.e., all 2^n states have same amplitude), and move it closer to (unknown) $|z\rangle$

Grover's Search

- Suppose f has a unique satisfying input z
  - Otherwise, modify f (by adding a hash "filter") so that with good probability it has a unique solution (if any)
- Plan: start with the uniform superposition on n-qubits (i.e., all 2^n states have same amplitude), and move it closer to (unknown) $|z\rangle$
  - Apply operations: (1) take $|x0\rangle$ to $|x\ f(x)\rangle$ (2) take $|x1\rangle$ to $-|x1\rangle$, and $|x0\rangle$ to $|x0\rangle$ and (3) take $|xy\rangle$ to $|x\ y\oplus f(x)\rangle$

Grover's Search

- Suppose f has a unique satisfying input z
  - Otherwise, modify f (by adding a hash "filter") so that with good probability it has a unique solution (if any)
- Plan: start with the uniform superposition on n-qubits (i.e., all 2^n states have same amplitude), and move it closer to (unknown) $|z\rangle$
  - Apply operations: (1) take $|x0\rangle$ to $|x\ f(x)\rangle$ (2) take $|x1\rangle$ to $-|x1\rangle$, and $|x0\rangle$ to $|x0\rangle$ and (3) take $|xy\rangle$ to $|x\ y\oplus f(x)\rangle$

uses scratch qubits

Grover's Search

- Suppose f has a unique satisfying input z
  - Otherwise, modify f (by adding a hash "filter") so that with good probability it has a unique solution (if any)
- Plan: start with the uniform superposition on n-qubits (i.e., all 2^n states have same amplitude), and move it closer to (unknown) $|z\rangle$
  - Apply operations: (1) take $|x0\rangle$ to $|x\ f(x)\rangle$ (2) take $|x1\rangle$ to $-|x1\rangle$, and $|x0\rangle$ to $|x0\rangle$ and (3) take $|xy\rangle$ to $|x\ y\oplus f(x)\rangle$
  - Takes $|z\rangle$ to $-|z\rangle$, and leaves other amplitudes unchanged

uses scratch qubits

Grover's Search

- Suppose f has a unique satisfying input z
  - Otherwise, modify f (by adding a hash "filter") so that with good probability it has a unique solution (if any)
- Plan: start with the uniform superposition on n-qubits (i.e., all 2^n states have same amplitude), and move it closer to (unknown) $|z\rangle$
  - Apply operations: (1) take $|x0\rangle$ to $|x\ f(x)\rangle$ (2) take $|x1\rangle$ to $-|x1\rangle$, and $|x0\rangle$ to $|x0\rangle$ and (3) take $|xy\rangle$ to $|x\ y{\oplus}f(x)\rangle$
  - Takes $|z\rangle$ to $-|z\rangle$, and leaves other amplitudes unchanged
  - One more "reflection" to take the vector close to $|z\rangle$

uses scratch qubits

Grover's Search

- Suppose f has a unique satisfying input z
  - Otherwise, modify f (by adding a hash "filter") so that with good probability it has a unique solution (if any)
- Plan: start with the uniform superposition on n-qubits (i.e., all 2^n states have same amplitude), and move it closer to (unknown) $|z\rangle$
  - Apply operations: (1) take $|x0\rangle$ to $|x\ f(x)\rangle$ (2) take $|x1\rangle$ to $-|x1\rangle$, and $|x0\rangle$ to $|x0\rangle$ and (3) take $|xy\rangle$ to $|x\ y \oplus f(x)\rangle$

uses scratch qubits

  - Takes $|z\rangle$ to $-|z\rangle$, and leaves other amplitudes unchanged
  - One more "reflection" to take the vector close to $|z\rangle$
- In $O(2^{n/2})$ iterations, amplitude of $|z\rangle$ becomes large (i.e., constant)

Shor’s Factoring

Shor’s Factoring

- By basic algebra, to factor a number N, enough to find the order r of a random number A (mod N)

Shor's Factoring

- By basic algebra, to factor a number N, enough to find the order r of a random number A (mod N)
  - i.e., smallest r s.t. $A^r \equiv 1 \pmod{N}$

Shor's Factoring

- By basic algebra, to factor a number N, enough to find the order r of a random number A (mod N)
  - i.e., smallest r s.t. $A^r \equiv 1 \pmod N$
- Prepare a superposition of states $|x\rangle\,|A^x \bmod N\rangle$ (for all x); pick random y_0 and make a measurement on second set of qubits to collapse the state to superposition over $|x\rangle|y_0\rangle$ where $x=x_0+ri$ (for all i)

Shor's Factoring

- By basic algebra, to factor a number N, enough to find the order r of a random number A (mod N)
  - i.e., smallest r s.t. $A^r \equiv 1 \pmod{N}$
- Prepare a superposition of states $|x\rangle\ |A^x \bmod N\rangle$ (for all x); pick random y_0 and make a measurement on second set of qubits to collapse the state to superposition over $|x\rangle|y_0\rangle$ where $x=x_0+ri$ (for all i)
  - Need to find the period r of this function

Shor's Factoring

- By basic algebra, to factor a number N, enough to find the order r of a random number A (mod N)
  - i.e., smallest r s.t. $A^r \equiv 1 \pmod{N}$
- Prepare a superposition of states $|x\rangle\ |A^x \bmod N\rangle$ (for all x); pick random y_0 and make a measurement on second set of qubits to collapse the state to superposition over $|x\rangle|y_0\rangle$ where $x=x_0+ri$ (for all i)
  - Need to find the period r of this function
  - Tool used: Quantum Fourier Transform

QFT for determining period

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \rightarrow \mathbb{C}$

QFT for determining period

- Recall Fourier Transform for functions $f: \{0,1\}^m \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)
- Fourier Transform of f: $\mathbb{Z}_M \rightarrow \mathbb{C}$

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)
- Fourier Transform of f: $\mathbb{Z}_M \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = \omega^{xy}$ (normalized), where $\omega = e^{i2\pi/M}$

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \to \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)
- Fourier Transform of f: $\mathbb{Z}_M \to \mathbb{C}$
  - Basis vectors: $X_x(y) = \omega^{xy}$ (normalized), where $\omega = e^{i2\pi/M}$
  - X_x is periodic (with period depending on x)

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)
- Fourier Transform of f: $\mathbb{Z}_M \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = \omega^{xy}$ (normalized), where $\omega = e^{i2\pi/M}$
  - X_x is periodic (with period depending on x)
  - If f is periodic, then $\dot{f}(x)$ (coefficient of X_x in f's FT) will be large for some x which is related to f's period

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)
- Fourier Transform of f: $\mathbb{Z}_M \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = \omega^{xy}$ (normalized), where $\omega = e^{i2\pi/M}$
  - X_x is periodic (with period depending on x)
  - If f is periodic, then $\dot{f}(x)$ (coefficient of X_x in f's FT) will be large for some x which is related to f's period
- QFT: initial state = $\Sigma_x\ f(x)\ |x\rangle$ and final state = $\Sigma_x\ \dot{f}(x)\ |x\rangle$

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \to \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)
- Fourier Transform of f: $\mathbb{Z}_M \to \mathbb{C}$
  - Basis vectors: $X_x(y) = \omega^{xy}$ (normalized), where $\omega = e^{i2\pi/M}$
  - X_x is periodic (with period depending on x)
  - If f is periodic, then $\dot{f}(x)$ (coefficient of X_x in f's FT) will be large for some x which is related to f's period
- QFT: initial state = Σ_x f(x) $|x\rangle$ and final state = Σ_x $\dot{f}(x)$ $|x\rangle$
  - Using an $O(\log^2 M)$ sized quantum circuit

QFT for determining period

- Recall Fourier Transform for functions f: $\{0,1\}^m \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = (-1)^{xy}$ (normalized)
- Fourier Transform of f: $\mathbb{Z}_M \rightarrow \mathbb{C}$
  - Basis vectors: $X_x(y) = \omega^{xy}$ (normalized), where $\omega = e^{i2\pi/M}$
  - X_x is periodic (with period depending on x)
  - If f is periodic, then $\dot{f}(x)$ (coefficient of X_x in f's FT) will be large for some x which is related to f's period
- QFT: initial state = $\Sigma_x\ f(x)\ |x\rangle$ and final state = $\Sigma_x\ \dot{f}(x)\ |x\rangle$
  - Using an $O(\log^2 M)$ sized quantum circuit
- Measuring the final state gives x with large coefficients with good probability. Enough to retrieve f's period.

Topics left out

Topics left out

- Derandomization and Extraction (lot of expander graphs here)

Topics left out

- Derandomization and Extraction (lot of expander graphs here)
- Hardness Amplification (useful in derandomization; lot of error correcting codes)

Topics left out

- Derandomization and Extraction (lot of expander graphs here)
- Hardness Amplification (useful in derandomization; lot of error correcting codes)
- More PCP and hardness of approximation (lot of Fourier analysis)

Topics left out

- Derandomization and Extraction (lot of expander graphs here)
- Hardness Amplification (useful in derandomization; lot of error correcting codes)
- More PCP and hardness of approximation (lot of Fourier analysis)
- More on Quantum Computation, Quantum error correction, Quantum communication (linear algebra over complex numbers)

Topics left out

- Derandomization and Extraction (lot of expander graphs here)
- Hardness Amplification (useful in derandomization; lot of error correcting codes)
- More PCP and hardness of approximation (lot of Fourier analysis)
- More on Quantum Computation, Quantum error correction, Quantum communication (linear algebra over complex numbers)
- Algebraic Models of Computation

Topics left out

- Derandomization and Extraction (lot of expander graphs here)
- Hardness Amplification (useful in derandomization; lot of error correcting codes)
- More PCP and hardness of approximation (lot of Fourier analysis)
- More on Quantum Computation, Quantum error correction, Quantum communication (linear algebra over complex numbers)
- Algebraic Models of Computation
- Logical characterizations, Proof complexity

Topics left out

- Derandomization and Extraction (lot of expander graphs here)
- Hardness Amplification (useful in derandomization; lot of error correcting codes)
- More PCP and hardness of approximation (lot of Fourier analysis)
- More on Quantum Computation, Quantum error correction, Quantum communication (linear algebra over complex numbers)
- Algebraic Models of Computation
- Logical characterizations, Proof complexity
- Cryptography...