Quantum algorithms I

Basic tools and algorithms

Jérémie Roland

Université Libre de Bruxelles

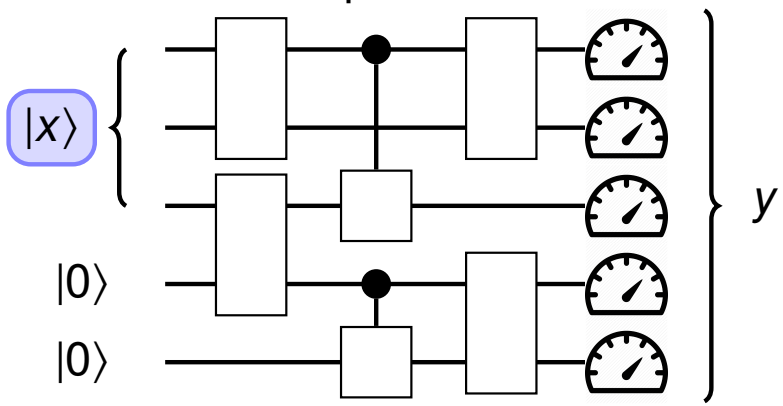
Quantum Information & Communication

Outline

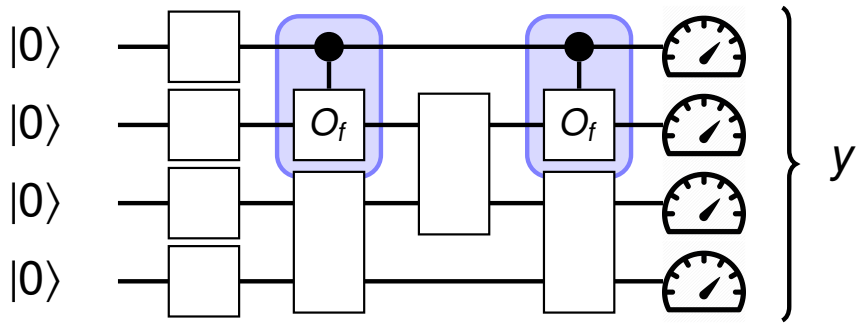
Outline

Oracle model

- In the usual circuit model
  - The input of the problem can be used as input to the circuit

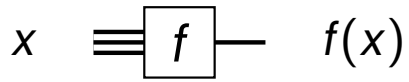


- In the oracle model
  - The input is only accessible via a black-box, called oracle

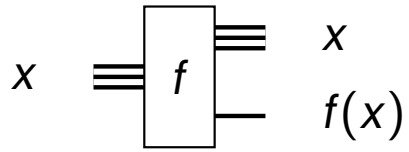


Oracles

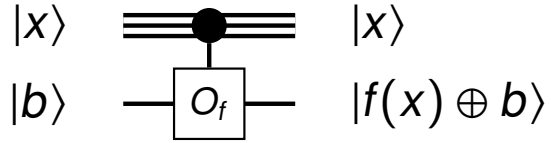
- The oracle computes an (unknown) function $f : \{0,1\}^n \to \{0,1\}$
  - Classical oracle



  - Reversible oracle



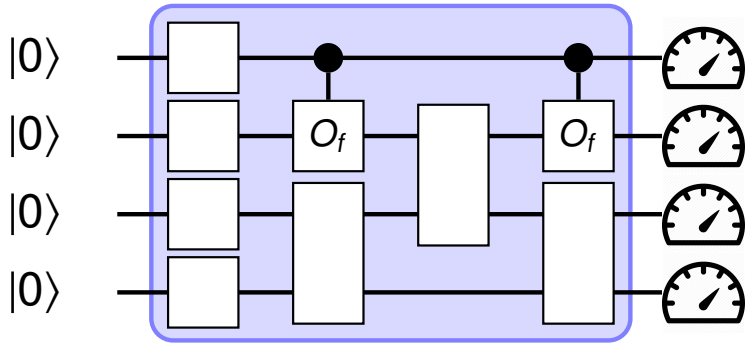
  - Quantum oracle



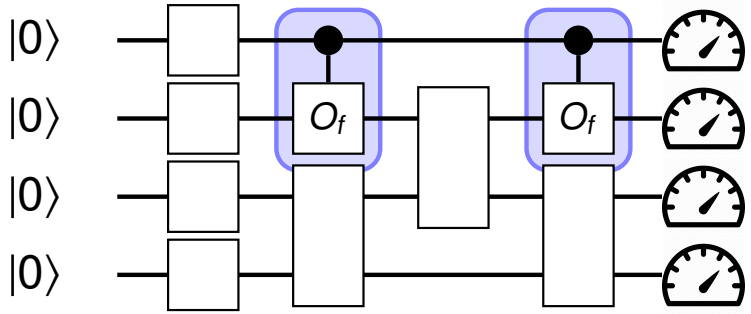
- The problem is to determine some property of f by querying the oracle

Time complexity vs query complexity

- Time complexity
  - Total number of gates: $T = 9$



- Query complexity
  - Number of oracle calls: $Q = 2$

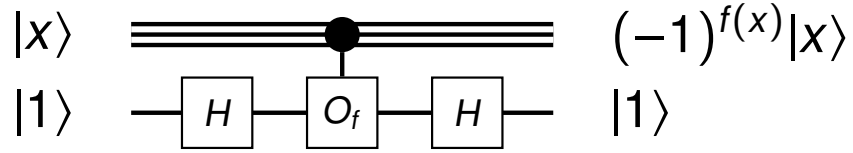


- Applications of query complexity
  - Lower bound on time complexity: $T \geq Q$
  - Equal up to constant factor for problems studied in these lectures

Outline

Tool: Phase kickback

- Let us consider the following circuit:



  - where H is the Hadamard gate

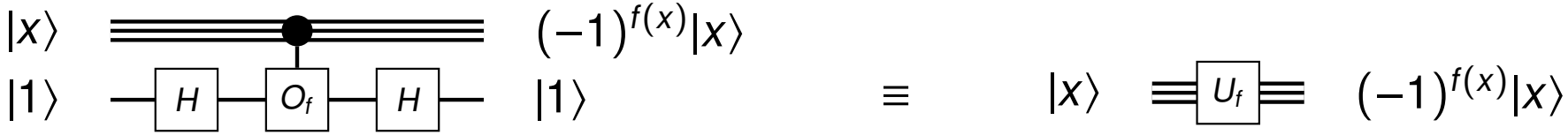
$$|0\rangle \xrightarrow{H} \tfrac{1}{\sqrt{2}}[|0\rangle + |1\rangle] \qquad |1\rangle \xrightarrow{H} \tfrac{1}{\sqrt{2}}[|0\rangle - |1\rangle]$$

- Analysis

$$\begin{aligned} |x\rangle|1\rangle &\xrightarrow{H} |x\rangle\tfrac{1}{\sqrt{2}}[|0\rangle - |1\rangle] \\ &\xrightarrow{O_f} |x\rangle\tfrac{1}{\sqrt{2}}[|f(x)\rangle - |f(x)\oplus 1\rangle] = (-1)^{f(x)}|x\rangle\tfrac{1}{\sqrt{2}}[|0\rangle - |1\rangle] \\ &\xrightarrow{H} (-1)^{f(x)}|x\rangle|1\rangle \end{aligned}$$

  - This computes $f(x)$ *in the phase*
  - $|1\rangle$ plays the role of an ancilla (returns to its initial state)

Tool: Phase kickback



Tool: Phase kickback

Using one call to oracle

$$O_f : |x\rangle|b\rangle \mapsto |x\rangle|f(x) \oplus b\rangle,$$

one can simulate the operation

$$U_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$$

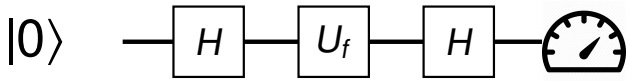
Application: Deutsch's algorithm [Deu85]

Definition of the problem

- Input: Black-box access to a one-bit function: $f : \{0,1\} \to \{0,1\}$
- Problem: Determine if $f(0) = f(1)$ or $f(0) \neq f(1)$

Deutsch's algorithm

- We use phase kickback



- Analysis

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$$

$$\xrightarrow{U_f} \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right] = (-1)^{f(0)}\frac{1}{\sqrt{2}}\left[|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle\right]$$

$$\xrightarrow{H} (-1)^{f(0)}|f(0)\oplus f(1)\rangle$$

  - The final measurement yields 0 if $f(0) = f(1)$, and 1 otherwise
  - This solves the problem with only one query, while two are required classically

Outline

Fourier transform over $\{0,1\}^n$

- Recall: H acts on one qubit $|x_1\rangle$ as

$$H|x_1\rangle = \frac{1}{\sqrt{2}}[|0\rangle + (-1)^{x_1}|1\rangle] = \frac{1}{\sqrt{2}} \sum_{y_1 \in \{0,1\}} (-1)^{x_1 y_1} |y_1\rangle$$

- Therefore, $H^{\otimes n}$ acts on an n-qubit state $|x\rangle = |x_1\rangle|x_2\rangle \dots |x_n\rangle$ as

$$|x\rangle \xrightarrow{H^{\otimes n}} \bigotimes_{i=1}^{n} H|x_i\rangle = \bigotimes_{i=1}^{n} \left[\frac{1}{\sqrt{2}} \sum_{y_i \in \{0,1\}} (-1)^{x_i y_i} |y_i\rangle \right] = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

  - where $x \cdot y = \sum_{i=1}^{n} x_i y_i$

- For a general n-qubit state, we have

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

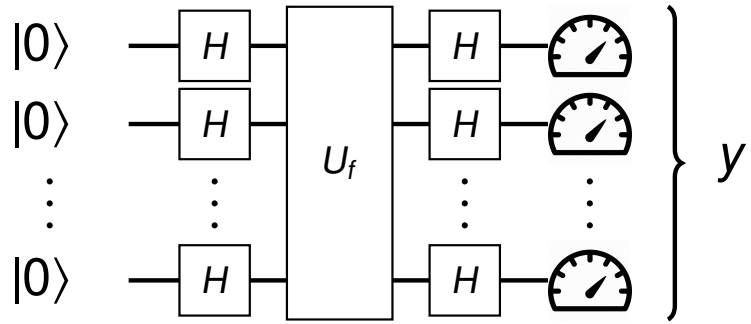
  - where

$$\hat{\psi}(y) = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot y} \psi(x)$$

  is the Fourier transform of $\psi(x)$.

Tool: Fourier sampling

- Let us consider the following circuit



- Analysis

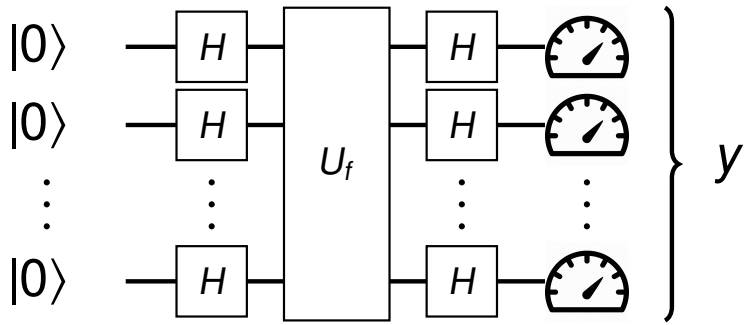
$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

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

$$\xrightarrow{H^{\otimes n}} \sum_{y\in\{0,1\}^n} \hat{\Phi}(y)|y\rangle$$

  - where $\Phi(x) = \frac{1}{\sqrt{2^n}}(-1)^{f(x)}$
  - The final measurement yields y with probability $p_y = |\hat{\Phi}(y)|^2$

Tool: Fourier sampling



Tool: Fourier sampling

Using one oracle call to f, one can sample y with probability p_y equal to the modulus square of the Fourier coefficient $\hat{\Phi}(y)$ of $\Phi(x) = \frac{1}{\sqrt{2^n}}(-1)^{f(x)}$

Note

- The Fourier coefficients $\hat{\Phi}(y)$ depend on all values $f(x)$ for all x.
- Fourier sampling can provide information about global properties of f

Application 1: Deutsch-Jozsa's algorithm [DJ92]

Definition of the problem

- Input: Black-box access to a (n-bit) function: $f : \{0,1\}^n \to \{0,1\}$
- Promise: $f(x)$ is either
  - constant: $f(x) = c \ \forall x$
  - balanced: $f(x) = 1$ on half of the values of x
- Problem: Determine if f is constant or balanced

Constant

x	000	001	010	011	100	101	110	111
$f(x)$	1	1	1	1	1	1	1	1

Balanced

x	000	001	010	011	100	101	110	111
$f(x)$	1	0	1	0	0	1	1	0

Application 1: Deutsch-Jozsa's algorithm [DJ92]

Constant

x	000	001	010	011	100	101	110	111
$f(x)$	1	1	1	1	1	1	1	1

Balanced

x	000	001	010	011	100	101	110	111
$f(x)$	1	0	1	0	0	1	1	0

Deutsch-Jozsa's algorithm

- We use Fourier sampling on f
- Analysis
  - We obtain y with probability $p_y = |\hat{\Phi}(y)|^2$, where $\hat{\Phi}(y) = \frac{1}{2^n} \sum_x (-1)^{f(x)+x\cdot y}$
  - If f is constant, that is, $f(x) = c$ for all x

$$\hat{\Phi}(y) = \tfrac{1}{2^n}(-1)^c \sum_{x\in\{0,1\}^n} (-1)^{x\cdot y} = \begin{cases} (-1)^c & \text{if } y = 0 \\ 0 & \text{otherwise} \end{cases}$$

  - If f is balanced

$$\hat{\Phi}(0) = \tfrac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} = 0$$

  - We always obtain $y = 0$ when f is constant
  - ... but never obtain $y = 0$ when f is balanced

Query complexity of the problem

Quantum query complexity

- Deutsch-Jozsa's algorithm solves the problem with one query

Classical query complexity

- We need $> 2^n/2$ queries to solve the problem with probability one
  - We could be unlucky and always obtain the same value $f(x)$ even if f is balanced
- We only need 2 queries to solve the problem with probability $2/3$
  - Query two random values x and x'
  - If $f(x) \neq f(x')$, conclude that f is balanced
  - If $f(x) = f(x')$, say that it is constant with probability $2/3$, and that it is balanced with probability $1/3$.

Application 2: Bernstein-Vazirani's algorithm [BV93]

Definition of the problem

- Input: Black-box access to a (n-bit) function: $f : \{0,1\}^n \to \{0,1\}$
- Promise: f is of the form $f(x) = a \odot x = \sum_i a_i x_i \mod 2$
- Problem: Find a

Bernstein-Vazirani's algorithm

- We use Fourier sampling on f
- Analysis
  - We obtain y with probability $p_y = |\hat{\Phi}(y)|^2$, where $\hat{\Phi}(y) = \frac{1}{2^n} \sum_x (-1)^{f(x)+x\cdot y}$
  - Since $f(x) = a \cdot x$, we have

$$\hat{\Phi}(y) = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{a\cdot x + x\cdot y} = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{(a\oplus y)\cdot x} = \begin{cases} 1 & \text{if } y = a \\ 0 & \text{otherwise} \end{cases}$$

  - We always obtain $y = a$

Query complexity of the problem

Quantum query complexity

- Bernstein-Vazirani's algorithm solves the problem with one query

Classical query complexity

- Each query reveals at most 1 bit of information about a
- a contains n bits, therefore we need $\Omega(n)$ queries to learn it
  - Even if we only require to learn it with constant probability

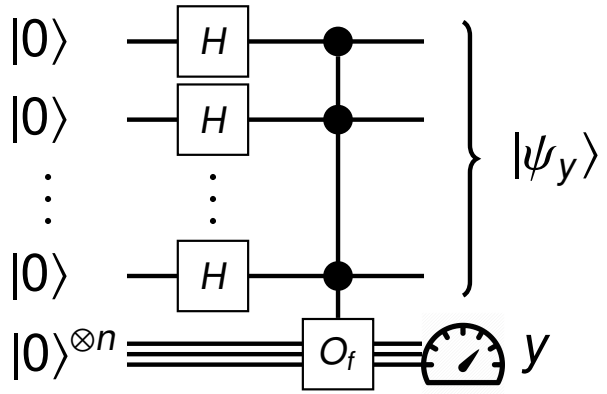
Note

- Linear separation between the quantum and classical complexities (1 vs $\Omega(n)$)
- The separation can be made exponential by using this construction recursively
  - Recursive Fourier sampling (see [BV93])
  - $BPP^O \neq BQP^O$ (see also next algorithm)

Outline

Tool: Preimage of a function

- Suppose we have black-box access to a function $f : \{0,1\}^n \to \{0,1\}^n$
- Let us consider the following circuit



  - Note that only the last n-qubit register is measured
- Analysis

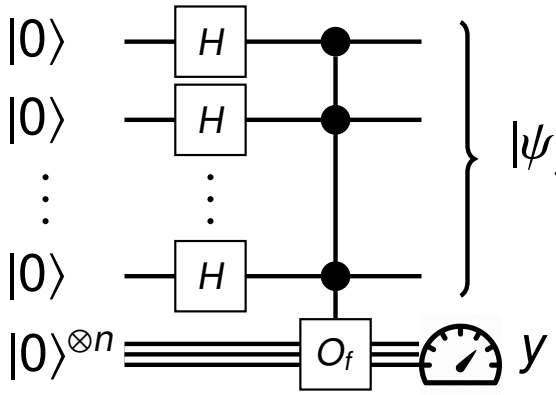
$$|0\rangle^{\otimes n}|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle^{\otimes n}$$

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

  - If the measurement outcome is y, then the remaining state is

$$|\psi_y\rangle = \frac{1}{\sqrt{|f^{-1}(y)|}} \sum_{x:f(x)=y} |x\rangle$$

Tool: Preimage of a function



$$|\psi_y\rangle = \frac{1}{\sqrt{|f^{-1}(y)|}} \sum_{x:f(x)=y} |x\rangle$$

Tool: Preimage of a function

Using one oracle call to f, one can generate a uniform superposition over all x's such that $f(x) = y$ (for some random value y).

Note

- The probability to obtain y is

$$p_y = \frac{|f^{-1}(y)|}{2^n}$$

  - where $f^{-1}(y) = \{x \in \{0,1\}^n : f(x) = y\}$

Application: Simon's algorithm [Sim94]

Definition of the problem

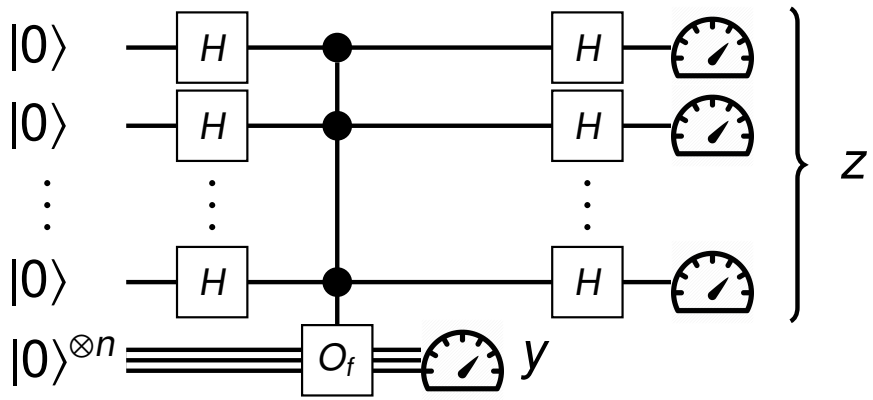
- Input: Black-box access to a function: $f : \{0,1\}^n \to \{0,1\}^n$
- Promise: f is *2-to-1* and *periodic* on $\{0,1\}^n$

$$f(x) = f(x') \quad \Leftrightarrow \quad x = x' \quad \text{or} \quad x = x' \oplus a$$

- Problem: Find the period a

Simon's algorithm

- We construct the preimage of f for some value y
- Then we apply a Fourier transform ($H^{\otimes n}$)



Simon's algorithm: analysis

- If the outcome of the first measurement is y, we have prepared

$$|\psi_y\rangle = \frac{1}{\sqrt{|f^{-1}(y)|}} \sum_{x:f(x)=y} |x\rangle = \frac{1}{\sqrt{2}} \left[|x_0\rangle + |x_0 \oplus a\rangle\right]$$

  - where x_0 is such that $f(x_0) = y$
- After the Fourier transform $H^{\otimes n}$, we have

$$\begin{aligned} |\psi_y\rangle \xrightarrow{H^{\otimes n}} & \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} \left[(-1)^{x_0\cdot z} + (-1)^{(x_0\oplus a)\cdot z}\right] |z\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} (-1)^{x_0\cdot z} \left[1 + (-1)^{a\cdot z}\right] |z\rangle \\ &= \frac{1}{\sqrt{2^{n-1}}} \sum_{z:a\odot z=0} (-1)^{x_0\cdot z} |z\rangle \end{aligned}$$

  - where $a \odot z = \sum_i a_i z_i \mod 2$
- The final measurement then produces outcome z uniformly at random from

$$\{z \in \{0,1\}^n : a \odot z = 0\}$$

Simon's algorithm: analysis (cont'd)

- How much information does a value z such that $a \odot z = 0$ provide about a?
  - Not that much (one parity bit), this just provides one linear equation satisfied by a

$$a_1 z_1 + a_2 z_2 + \ldots a_n z_n = 0 \mod 2$$

- By repeating the procedure, we can obtain multiple values $z^{(1)}, z^{(2)}, \ldots, z^{(n)}$
  - ... and form a linear system of equations that can be solved for a

$$\begin{cases} a_1 z_1^{(1)} + a_2 z_2^{(1)} + \ldots + a_n z_n^{(1)} = 0 \mod 2 \\ a_1 z_1^{(2)} + a_2 z_2^{(2)} + \ldots + a_n z_n^{(2)} = 0 \mod 2 \\ \vdots \\ a_1 z_1^{(n)} + a_2 z_2^{(n)} + \ldots + a_n z_n^{(n)} = 0 \mod 2 \end{cases}$$

- It suffices to repeat $O(n)$ times in order to obtain n linearly independent equations
  - This follows from the fact that z is uniform in $\{z \in \{0,1\}^n : a \odot z = 0\}$.

Query complexity of the problem

Quantum query complexity

- Simon's algorithm solves the problem with $O(n)$ queries

Classical query complexity

- Queries reveal essentially nothing about a until we find a pair x, x' such that $x' = x \oplus a$
- For one random pair x, x', we have $\Pr[x' = x \oplus a] = 2^{-n}$
- For T queries, we have less than T^2 pairs, so that $\Pr[\text{find a}] \leq T^2 2^{-n}$
- Therefore, to find a with probability p, we need $T \geq p2^{n/2}$

Notes

- Exponential quantum speed-up $O(n)$ vs $\Omega(2^{n/2})$
  - Separation between BQP^f and BPP^f
- In the algorithm, we never used the measurement outcome y
  - We don't need to actually perform the measurement (only useful for analysis)

Tomorrow

Last lecture

- Grover's algorithm
- Amplitude amplification
- Shor's algorithm

References I

[BV93] Ethan Bernstein and Umesh Vazirani.
Quantum complexity theory.
In Proc. 25th STOC, pages 11–20, 1993.

[Deu85] David Deutsch.
Quantum theory, the Church-Turing principle and the universal quantum computer.
Proc. R. Soc. London A, 400:97–117, 1985.

[DJ92] David Deutsch and Richard Jozsa.
Rapid solution of problems by quantum computation.
Proc. R. Soc. London A, 439:553, 1992.

[Sim94] Daniel R. Simon.
On the power of quantum computation.
In Proc. 35th FOCS, pages 116–123, 1994.