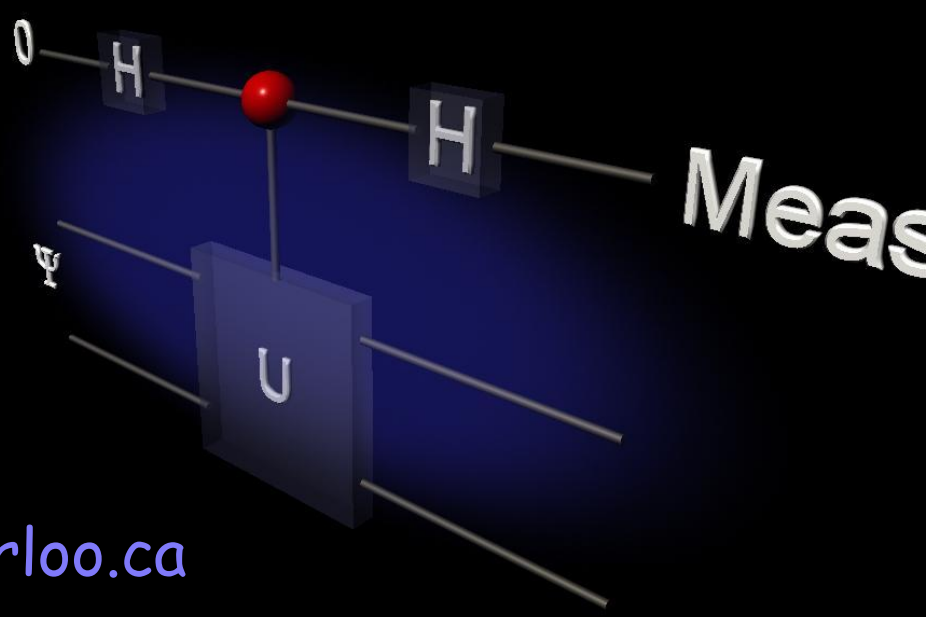
Introduction to Quantum Information Processing



CO481 CS467 PHYS467

Michele Mosca mmosca@iqc.uwaterloo.ca

Tuesdays and Thursdays 10am-11:15am

IQC Institute for Quantum Computing

UNIVERSITY OF WATERLOO

PI PERIMETER INSTITUTE FOR THEORETICAL PHYSICS

Overview

Lecture 9

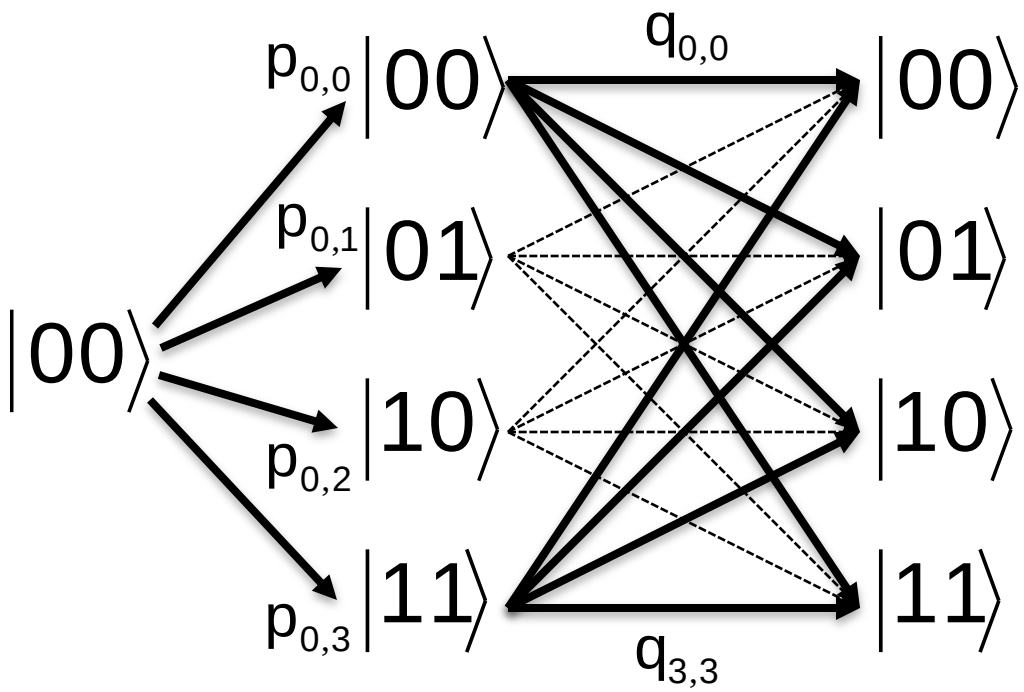
Michele Mosca
(borrowing some overheads from Richard Cleve)

- Introduction to quantum algorithms
- Parity problem and Deutsch's algorithm
- Constant vs. balanced problem
- Computing H⊗H⊗ ... ⊗H
- Simon's problem

- **Introduction to quantum algorithms**
- Parity problem and Deutsch's algorithm
- Constant vs. balanced problem
- Computing H⊗H⊗ ... ⊗H
- Simon's problem

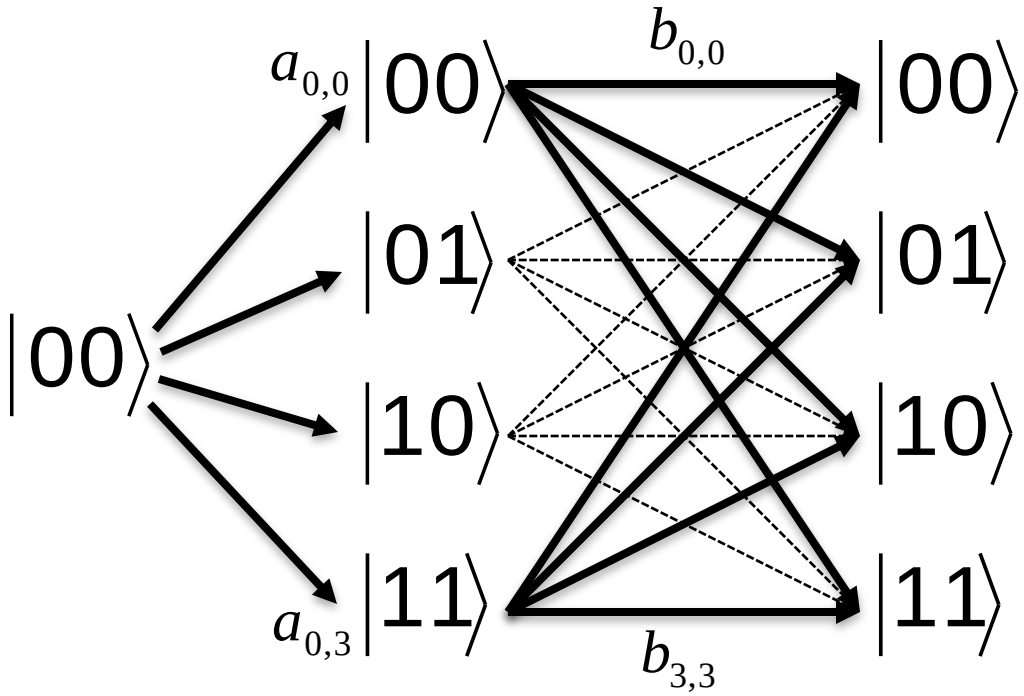
A classical randomized algorithm

- Several computational paths leading to the same outcome.
- Add up the probabilities.



$$\Pr(00) = \sum_j p_{0,j} q_{j,0}$$

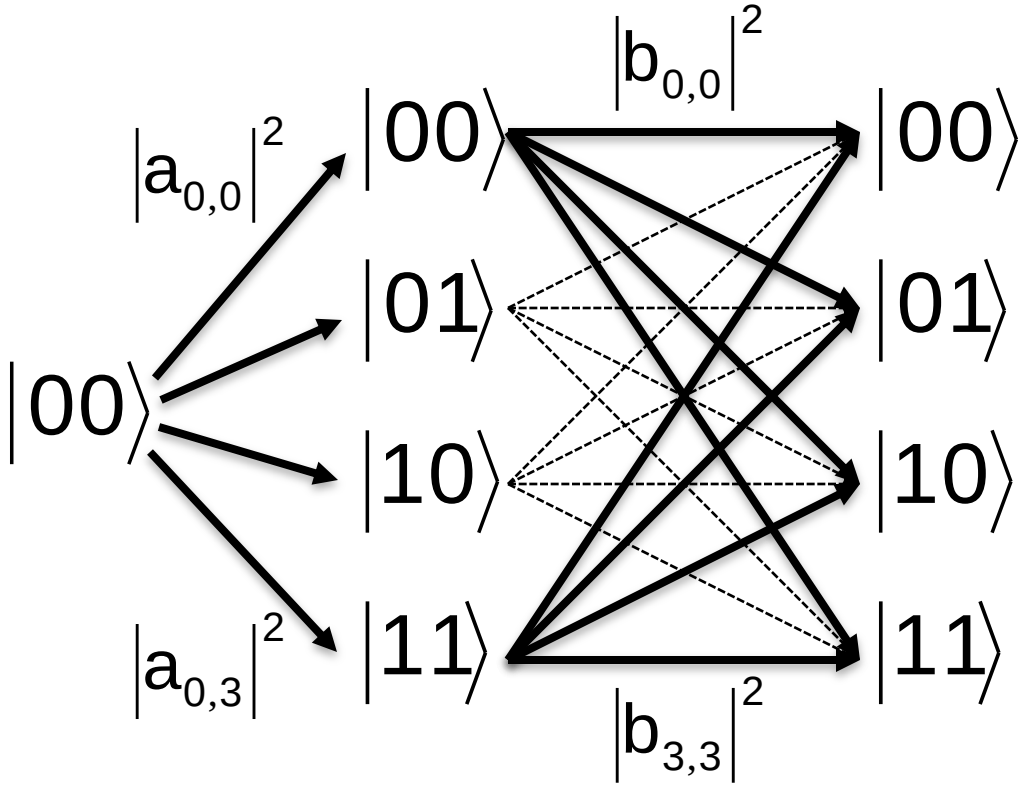
A quantum algorithm



$$\Pr(00) = \left| \sum_j a_{0,j} b_{j,0} \right|^2$$

What if we measure along the way?

If we look at the state of the system at each step, it behaves like a classical randomized algorithm.



$$\Pr(00) = \sum_j |a_{0,j}|^2 |b_{j,0}|^2$$

$$= \sum_j |a_{0,j} b_{j,0}|^2$$

Decoherence

- A quantum system that is continually measured (or “leaks” information to an external system) will behave like a classical randomized system.
- Partial measurements will give a probability distribution somewhere in between the two extremes.
- Error-correcting codes will allow a quantum system interacting with the environment to maintain “coherence”.

How do quantum algorithms work?

Given a polynomial-time classical algorithm for $f:\{0,1\}^n \longrightarrow T$, it is straightforward to construct a quantum algorithm that creates the state

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$$

at the cost of about ***one*** evaluation of f

Is this exponentially many computations at polynomial cost?

No! — the most straightforward way of extracting information from the state yields just $(x, f(x))$ for a random $x \in \{0,1\}^n$

<u>But we can make some interesting ***tradeoffs***</u>:

instead of learning about any $(x, f(x))$ point, one can learn something about a ***global property*** of f

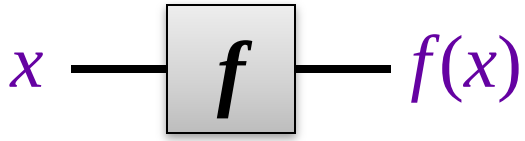
Quantum algorithms

- Quantum Algorithms should exploit quantum parallelism *and* quantum interference.

- This is necessary, but not sufficient, in order to outperform a classical probabilistic algorithm. E.g. at some point in the execution of the algorithm, the state of the system should have a substantial amount of entanglement (assuming we are in the usual model of unitary operations on pure states).

Query scenario

Input: a function f, given as a black box (a.k.a. oracle)



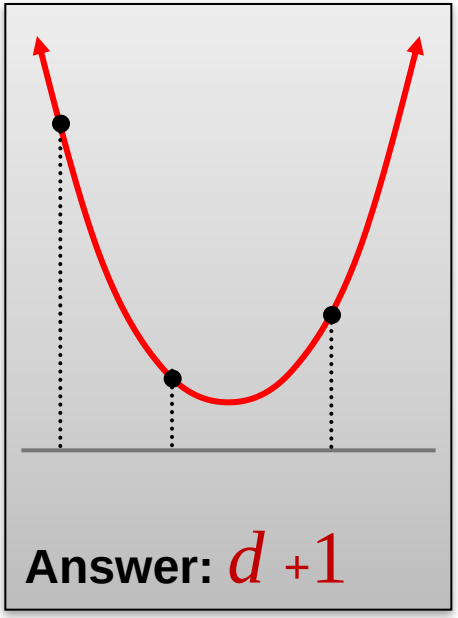
Goal: determine some information about f making as few queries to f as possible (of course, other operations are allowed – but we do not count them)

Example: polynomial interpolation

Let: $f(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_d x^d$

Goal: determine $c_0, c_1, c_2, \ldots, c_d$

Question: How many classical f-queries does one require for this?



- Introduction to quantum algorithms
- **Parity problem and Deutsch's algorithm**
- Constant vs. balanced problem
- Computing H⊗H⊗ ... ⊗H
- Simon's problem

Deutsch's problem

Let $f : \{0,1\} \rightarrow \{0,1\}$

There are ***four*** possibilities:

x	$f_1(x)$
0	0
1	0

x	$f_2(x)$
0	1
1	1

x	$f_3(x)$
0	0
1	1

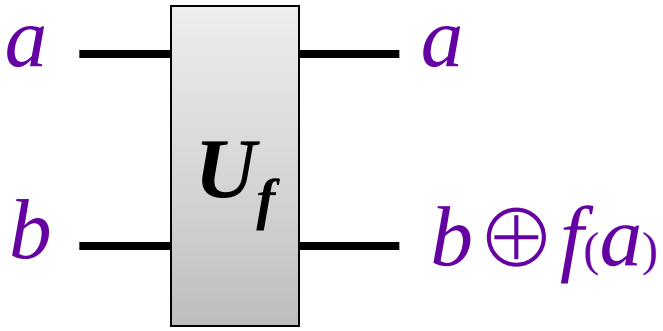
x	$f_4(x)$
0	1
1	0

Goal: determine whether or not $f(0) = f(1)$ (i.e. $f(0) \oplus f(1)$)

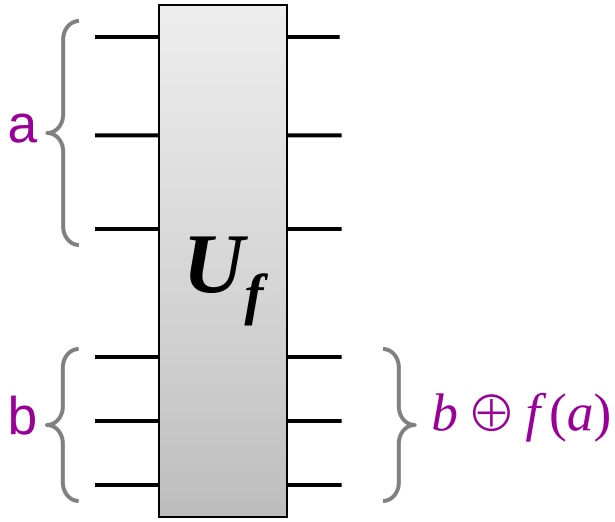
Any classical method requires ***two*** queries

What about a quantum method?

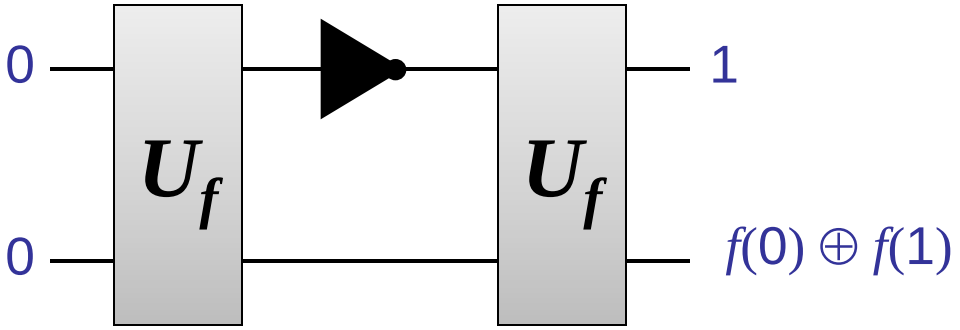
Unitary black box for *f*



a and b can be more than one qubit



A classical algorithm:
(still requires 2 queries)



2 queries + **1** auxiliary operation

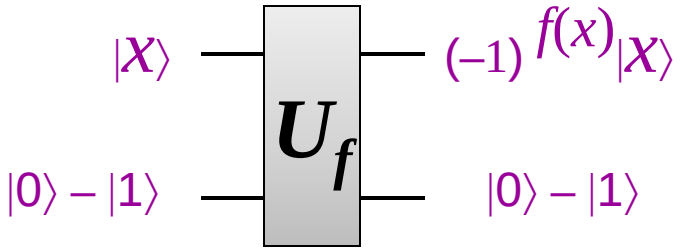
Quantum algorithm (1)

Is there some way to construct $\frac{(-1)^{f(0)}}{\sqrt{2}}|0\rangle + \frac{(-1)^{f(1)}}{\sqrt{2}}|1\rangle$?

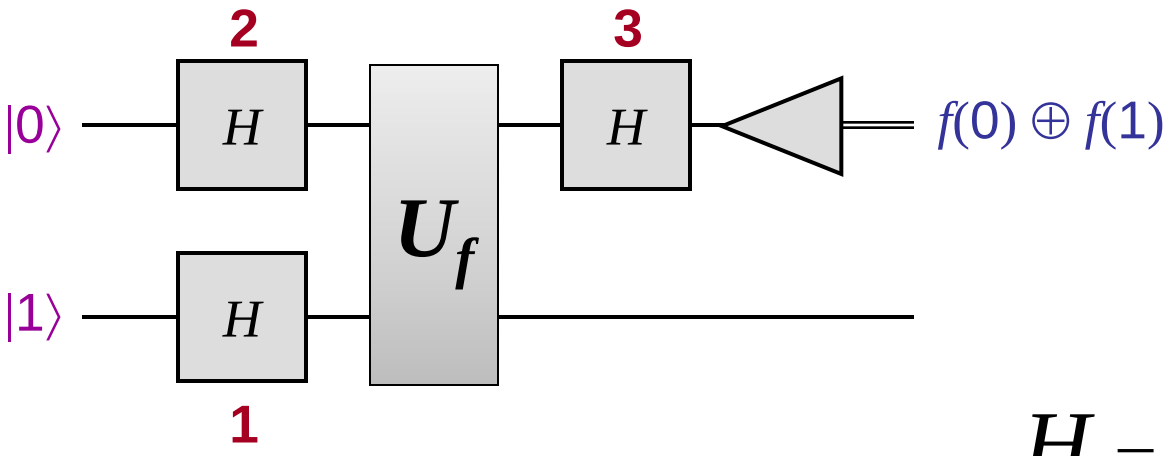
$$= (-1)^{f(0)}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^{f(0)\oplus f(1)}}{\sqrt{2}}|1\rangle\right)$$

Why would I want that?

Note how we can use U_f to induce a ***phase shift*** of $(-1)^{f(x)}$ to $|x\rangle$



Quantum algorithm for Deutsch



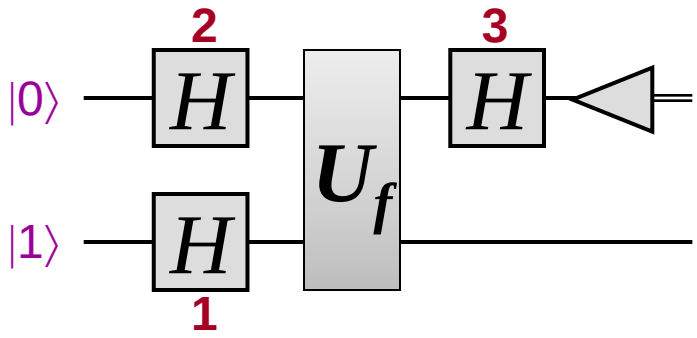
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

1 query + **4** auxiliary operations

How does this algorithm work?

Each of the three H operations can be seen as playing a different role ...

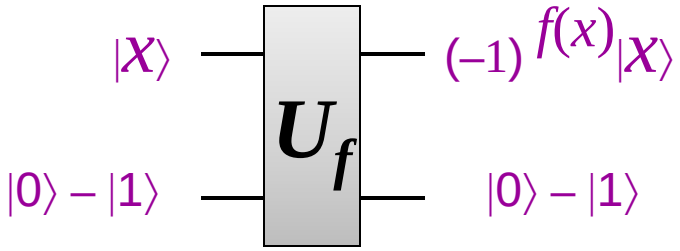
Quantum algorithm (1)



1. Creates the state $|0\rangle - |1\rangle$, which is an eigenvector of

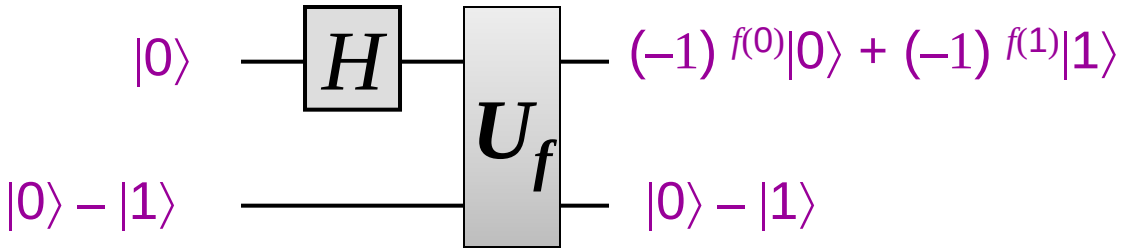
- **NOT** with eigenvalue –1
- I with eigenvalue +1

This causes f to induce a ***phase shift*** of $(-1)^{f(x)}$ to $|x\rangle$



Quantum algorithm (2)

2. Causes f to be queried ***in superposition*** (at $|0\rangle + |1\rangle$)



x	$f_1(x)$
0	0
1	0

x	$f_2(x)$
0	1
1	1

$\pm(|0\rangle + |1\rangle)$

x	$f_3(x)$
0	0
1	1

x	$f_4(x)$
0	1
1	0

$\pm(|0\rangle - |1\rangle)$

Quantum algorithm (3)

3. Distinguishes between $\pm(|0\rangle + |1\rangle)$ and $\pm(|0\rangle - |1\rangle)$

$$\pm(|0\rangle + |1\rangle) \xleftrightarrow{H} \pm|0\rangle$$

$$\pm(|0\rangle - |1\rangle) \xleftrightarrow{H} \pm|1\rangle$$

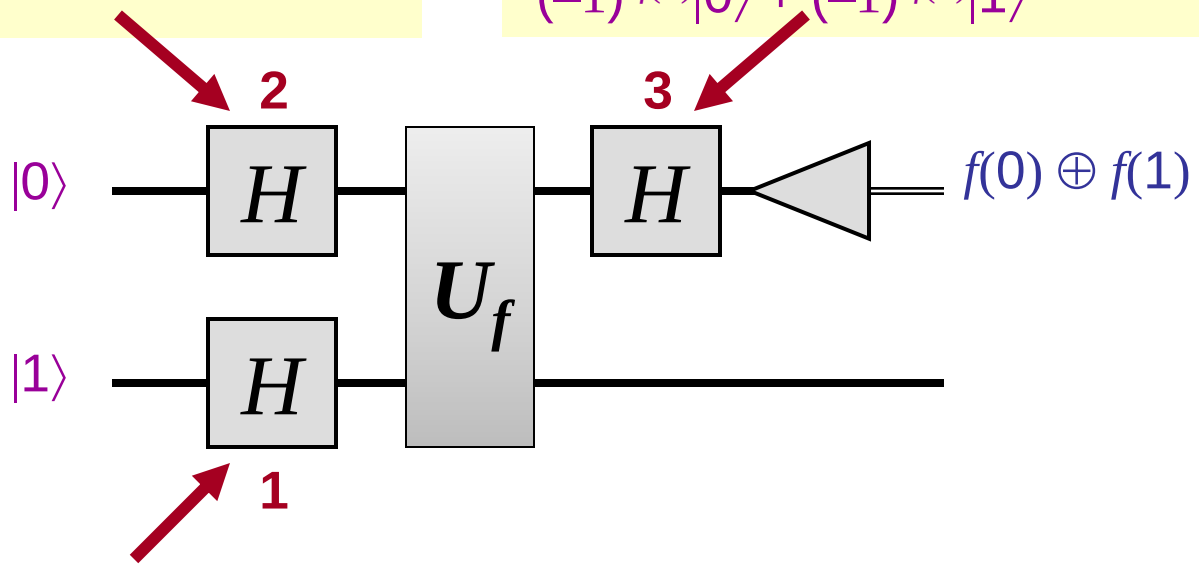
Summary of Deutsch's algorithm

Makes only one query, whereas two are needed classically

produces superpositions of inputs to f : $|0\rangle + |1\rangle$

extracts phase differences from

$(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle$



constructs eigenvector so f-queries induce phases: $|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$

- Introduction to quantum algorithms
- Parity problem and Deutsch's algorithm
- **Constant vs. balanced problem**
- Computing H⊗H⊗ ... ⊗H
- Simon's problem

Constant vs. balanced

Let $f: \{0,1\}^n \rightarrow \{0,1\}$ be either constant or balanced, where

- ***constant*** means $f(x) = 0$ for all x, or $f(x) = 1$ for all x
- ***balanced*** means $\Sigma_x \ f(x) = \ 2^{n-1}$

Goal: determine whether f is constant or balanced

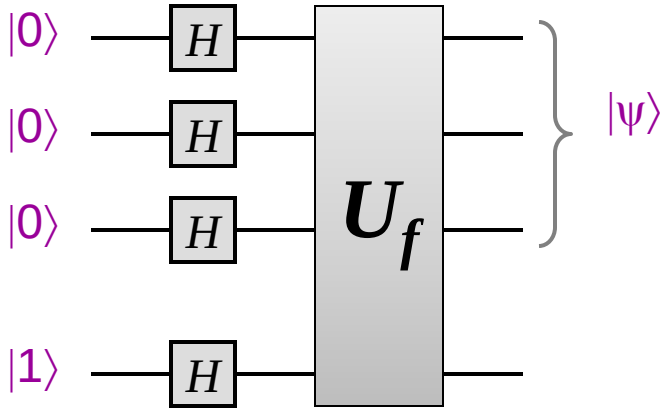
How many queries are there needed *classically?* $2^{n-1}+1$

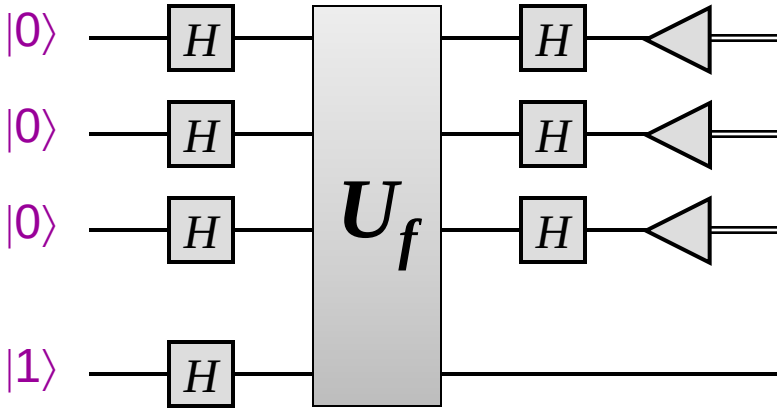
Example: if $f(0000) = f(0001) = f(0010) = \ldots = f(0111) = 0$ then it still could be either

Quantumly? just 1 query suffices!

[Deutsch & Jozsa, 1992]

Quantum algorithm





Constant case: $|\psi\rangle = \pm \sum_x |X\rangle$ ***Why?***

Balanced case: $|\psi\rangle$ is ***orthogonal*** to $\pm \sum_x |X\rangle$ ***Why?***

How to distinguish between the cases? What is $H^{\otimes n}|\psi\rangle$?

Constant case: $H^{\otimes n}|\psi\rangle = \pm |00...0\rangle$

Balanced case: $H^{\otimes n}|\psi\rangle$ is orthogonal to $|0...00\rangle$

Last step of the algorithm: if the measured result is 000 then output “constant”, otherwise output “balanced”

Probabilistic *classical* algorithm solving constant vs. balanced

But here's a classical procedure that makes only **2** queries and performs fairly well probabilistically:

1. pick $x_1, x_2 \in \{0,1\}^n$ randomly
2. **if** $f(x_1) \neq f(x_2)$ **then** output balanced **else** output constant

What happens if f is constant? The algorithm always succeeds

What happens if f is balanced? Succeeds with probability ½

Sampling k times gives one-sided error probability that decays exponentially in k.

Therefore, for large n, << 2^n queries are likely sufficient.

One class of "balanced" or constant functions: f(x) = a·x for $a \in \{0,1\}^n$; Bernstein-Vazirani algorithm finds a.

- Introduction to quantum algorithms
- Parity problem and Deutsch's algorithm
- Constant vs. balanced problem
- **Computing H⊗H⊗ ... ⊗H**
- Simon's problem

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

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

where $x \cdot y = x_1 y_1 \oplus \ldots \oplus x_n y_n$

Example: $$H \otimes H = \frac{1}{2} \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

Proof: For all $x \in \{0,1\}$, $H|x\rangle = |0\rangle + (-1)^x |1\rangle = \Sigma_y (-1)^{xy} |y\rangle$

Thus, $H^{\otimes n}|x_1 \ldots x_n\rangle = (\Sigma_{y_1} (-1)^{x1y1}|y_1\rangle) \ldots (\Sigma_{y_n} (-1)^{xnyn}|y_n\rangle)$

$= \Sigma_y (-1)^{x1y1 \oplus \ldots \oplus xnyn} |y_1 \ldots y_n\rangle$ ∎

- Introduction to quantum algorithms
- Parity problem and Deutsch's algorithm
- Constant vs. balanced problem
- Computing H⊗H⊗ ... ⊗H
- **Simon's problem**

Quantum vs. classical separations

Black-box problem	Quantum	Classical	
Deutsch's problem	**1** (query)	**2** (queries)	
constant vs. balanced	**1**	$\frac{1}{2}\,2^n + 1$	**(only for exact)**
Bernstein-Vazirani problem	**1**	**n**	
Simon's problem	$O(n)$	$\theta(2^{n/2})$	**(probabilistic)**

Simon's problem

Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ have the property that there exists an $s \in \{0,1\}^n$ such that $f(x) = f(y)$ iff $x \oplus y = s$ or $x = y$

Example:

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

What is **s** is this case? s= **101**

A classical algorithm for Simon

Search for a ***collision***, an $x \neq y$ such that $f(x) = f(y)$

1. Choose $x_1, x_2, ..., x_k \in \{0,1\}^n$ uniformly randomly (independently)

2. For all $i \neq j$, if $f(x_i) = f(x_j)$ then output $x_i \oplus x_j$ and halt

A hard case is where s is chosen randomly from $\{0,1\}^n - \{0^n\}$ and then the "table" for f is filled out randomly subject to the structure implied by s

Question: How big does k have to be for the probability of a collision to be a constant, such as ¾?

Answer: order $2^{n/2}$

Classical lower bound

Theorem: ***any*** classical algorithm solving Simon's problem must make $\Omega(2^{n/2})$ queries

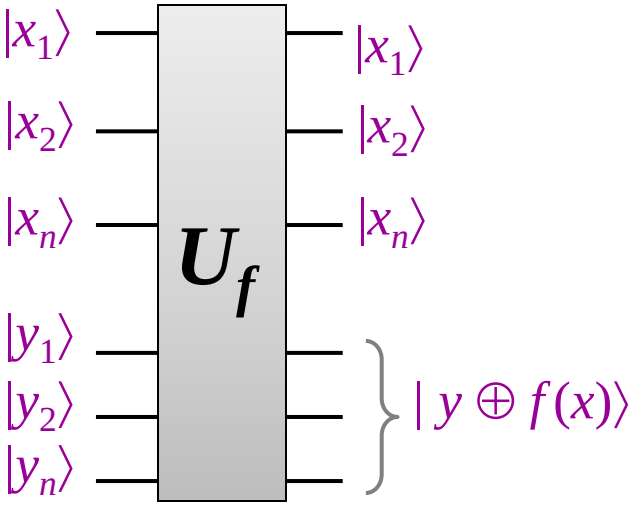
Proof is omitted here

Note: the performance analysis of the previous algorithm does ***not*** imply the theorem

… how can we know that there isn't a ***different*** algorithm that performs better?

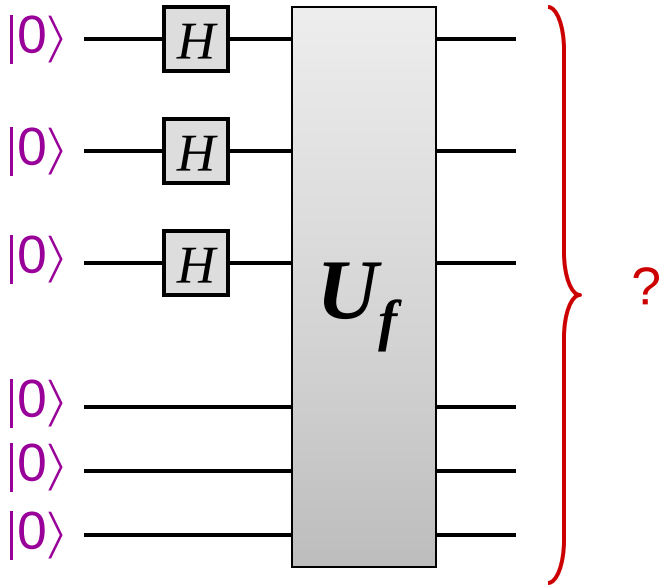
A quantum algorithm for Simon (1)

Queries:



Proposed start of quantum algorithm: query all values of f in superposition

What is the output state of this circuit?



A quantum algorithm for Simon (2)

Answer: the output state is $\sum_{x\in\{0,1\}^n} |x\rangle |f(x)\rangle$

Let $T \subseteq \{0,1\}^n$ be such that ***one*** element from each matched pair is in T (assume $s \neq 00...0$)

Example: could take T = {000, 001, 011, 111}

Then the output state can be written as:

$$\sum_{x\in T} |x\rangle |f(x)\rangle + |x\oplus s\rangle |f(x\oplus s)\rangle$$

$$= \sum_{x\in T} \left(|x\rangle + |x\oplus s\rangle\right) |f(x)\rangle$$

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

A quantum algorithm for Simon (3)

Measuring the second register yields $|x\rangle + |x \oplus s\rangle$ in the first register, for a random $x \in T$

How can we use this to obtain ***some*** information about s ?

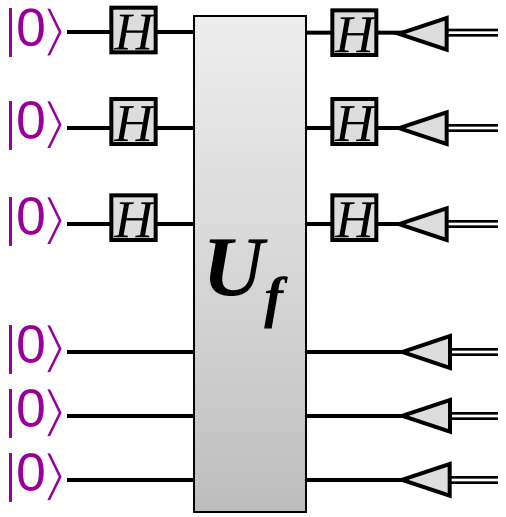
Try applying $H^{\otimes n}$ to the state, yielding:

$$\sum_{y\in\{0,1\}^n}(-1)^{x\bullet y}|y\rangle + \sum_{y\in\{0,1\}^n}(-1)^{(x\oplus s)\bullet y}|y\rangle$$

$$= \sum_{y\in\{0,1\}^n}(-1)^{x\bullet y}\left(1+(-1)^{s\bullet y}\right)|y\rangle$$

Measuring this state yields y with prob. $\begin{cases} (1/2)^{n-1} & \text{if } s\cdot y = 0 \\ 0 & \text{if } s\cdot y \neq 0 \end{cases}$

A quantum algorithm for Simon (4)



Executing this algorithm $k = O(n)$ times yields random $y_1, y_2, ..., y_k \in \{0,1\}^n$ such that $s \cdot y_1 = s \cdot y_2 = ... = s \cdot y_n = 0$

How does this help?

This is a system of k linear equations:

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{kn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

With high probability, there is a unique non-zero solution that is s (which can be efficiently found by linear algebra)

Conclusion of Simon's algorithm

- Any classical algorithm has to query the black box $\Omega(2^{n/2})$ times, even to succeed with probability ¾.

- There is a quantum algorithm that queries the black box only $O(n)$ times, performs only $O(n^3)$ auxiliary operations (for the Hadamards, measurements, and linear algebra), and succeeds with probability ¾.