

**Introduction to  
Quantum Information Processing  
QIC 710 / CS 768 / PH 767 / CO 681 / AM 871**

**Lecture 4 (2019)**

**Richard Cleve**

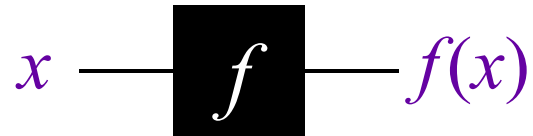
QNC 3129

[cleve@uwaterloo.ca](mailto:cleve@uwaterloo.ca)

# Simple quantum algorithms in the query scenario

# Query scenario

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



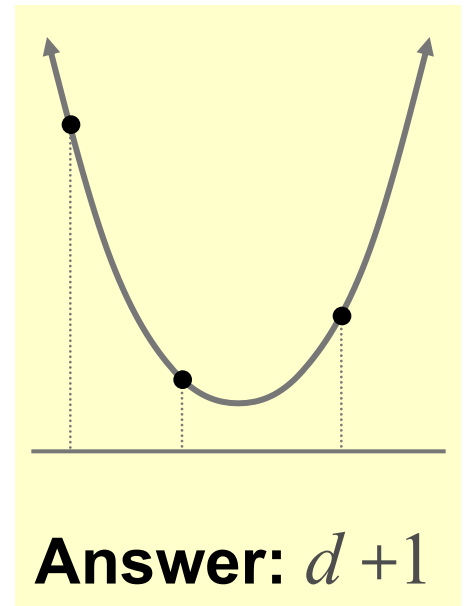
**Goal:** determine some information about  $f$  making as few queries to  $f$  (and other operations) as possible

**Example:** polynomial interpolation

**Let:**  $f(x) = c_0 + c_1x + c_2x^2 + \dots + c_dx^d$

**Goal:** determine  $c_0, c_1, c_2, \dots, c_d$

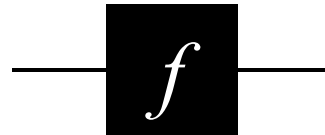
**Question:** How many  $f$ -queries does one require for this?



# Deutsch'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

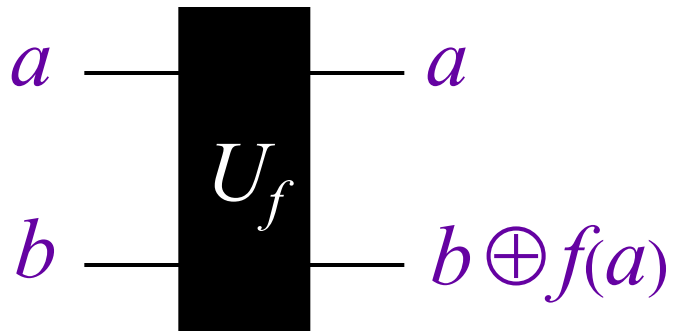
$x$	$f_4(x)$
0	1
1	0

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

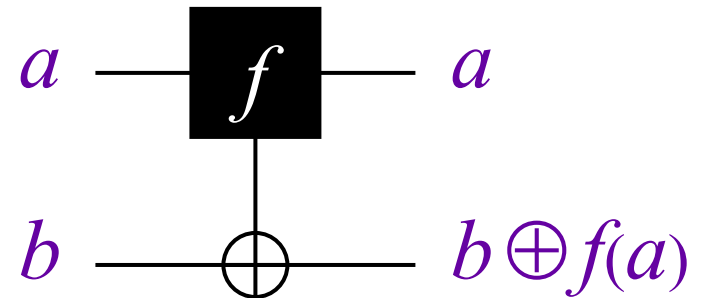
Any classical method requires **two** queries

What about a quantum method?

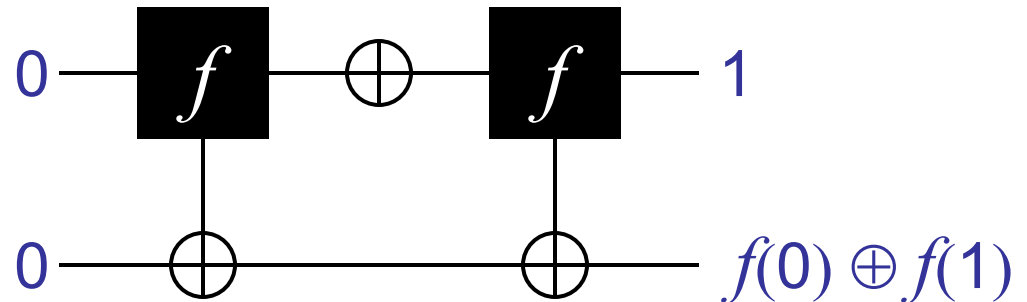
# Reversible black box for $f$



alternate  
notation:

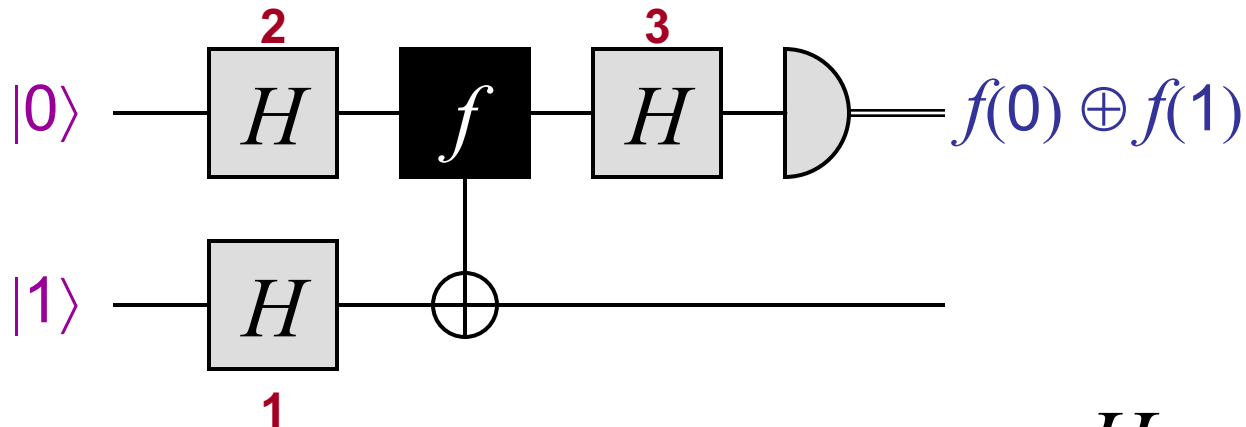


**A classical algorithm:**  
(still requires 2 queries)



**2 queries + 1 auxiliary operation**

# Quantum algorithm for Deutsch



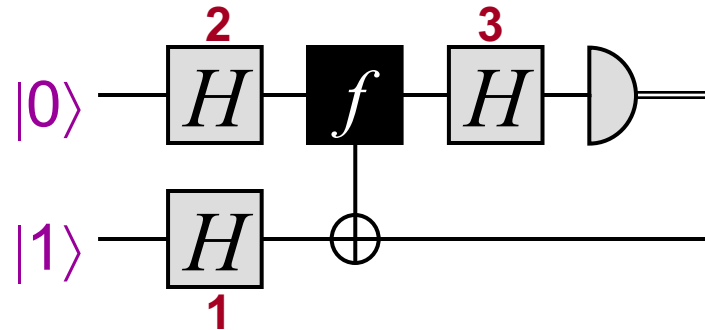
1 query + 4 auxiliary operations

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

How does this algorithm work?

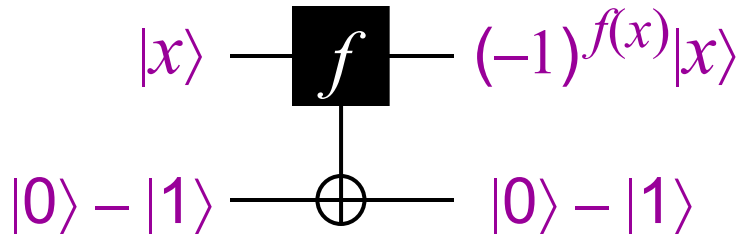
Each of the three  $H$  operations can be seen as playing a different role ...

# Quantum algorithm (1)



- Creates the state  $|0\rangle - |1\rangle$ , which is an eigenvector of
 
$$\begin{cases} \text{NOT with eigenvalue } -1 \\ \mathbf{I} \quad \text{with eigenvalue } +1 \end{cases}$$

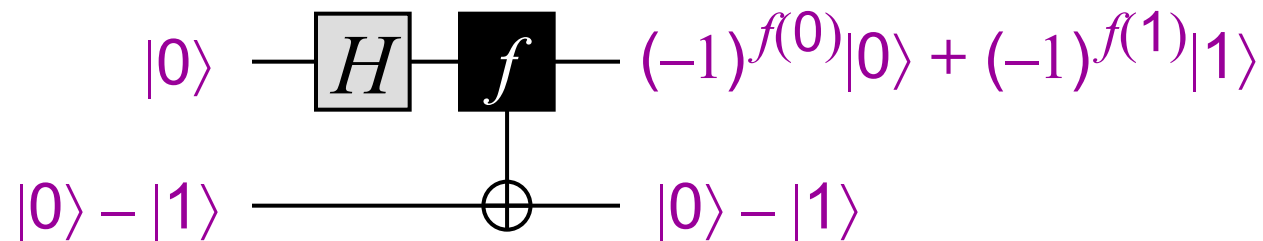
This causes  $f$  to induce a **phase shift** of  $(-1)^{f(x)}$  to  $|x\rangle$





# 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

$x$	$f_3(x)$
0	0
1	1

$x$	$f_4(x)$
0	1
1	0

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

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

# Quantum algorithm (3)

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

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

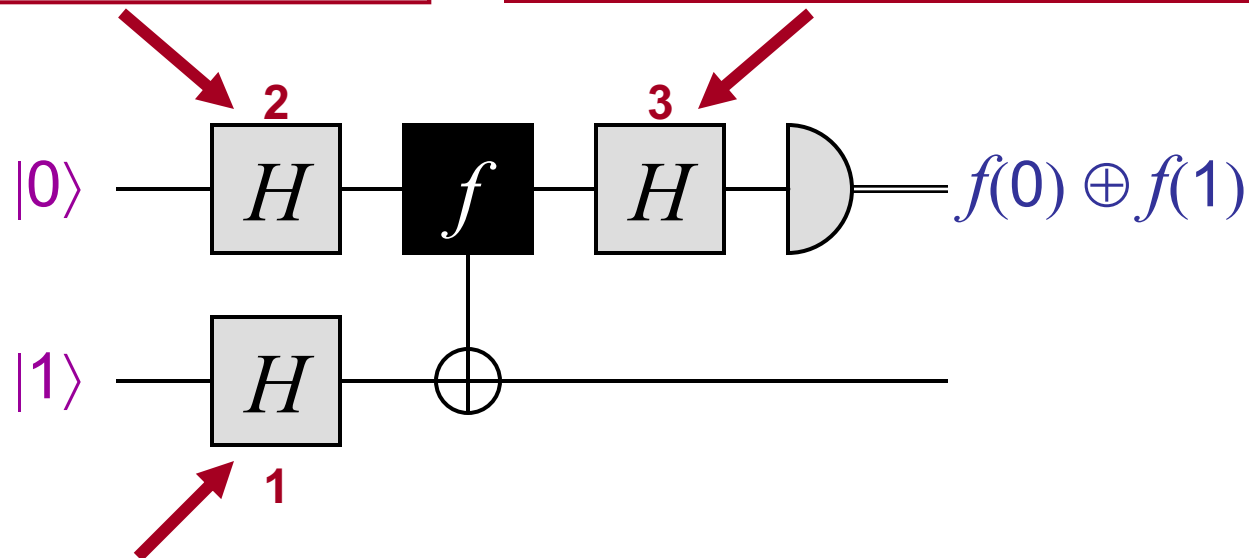
$$\pm(|0\rangle - |1\rangle) \xleftrightarrow{H} \pm|1\rangle$$

# 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:  $|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$

# One-out-of-four search

# One-out-of-four search

Let  $f: \{0,1\}^2 \rightarrow \{0,1\}$  have the property that there is exactly one  $x \in \{0,1\}^2$  for which  $f(x) = 1$

Four possibilities:

$x$	$f_{00}(x)$	$x$	$f_{01}(x)$	$x$	$f_{10}(x)$	$x$	$f_{11}(x)$
00	1	00	0	00	0	00	0
01	0	01	1	01	0	01	0
10	0	10	0	10	1	10	0
11	0	11	0	11	0	11	1

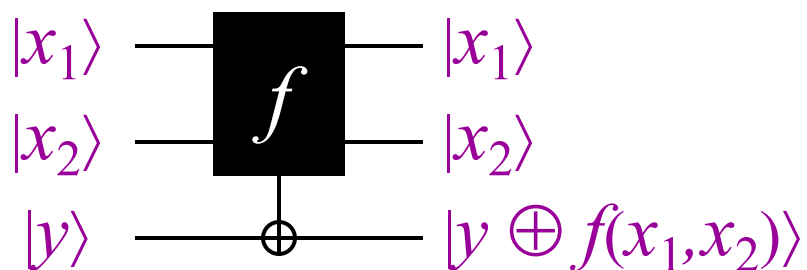
**Goal:** find  $x \in \{0,1\}^2$  for which  $f(x) = 1$

What is the minimum number of queries **classically?** \_\_\_\_\_

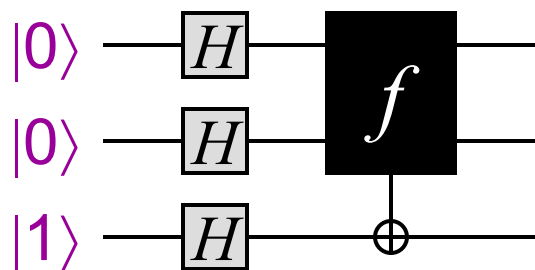
**Quantumly?** \_\_\_\_\_

# Quantum algorithm (I)

Black box for 1-4 search:



Start by creating phases in superposition of all inputs to  $f$ :



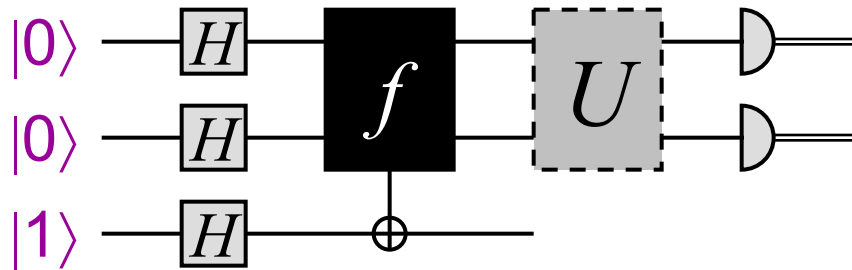
**Input** state to query?

$$(|00\rangle + |01\rangle + |10\rangle + |11\rangle)(|0\rangle - |1\rangle)$$

**Output** state of query?

$$((-1)^{f(00)}|00\rangle + (-1)^{f(01)}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle)(|0\rangle - |1\rangle)$$

# Quantum algorithm (II)



$\leftarrow$  Apply the  $U$  that maps  
 $\leftarrow |\psi_{00}\rangle, |\psi_{01}\rangle, |\psi_{10}\rangle, |\psi_{11}\rangle$  to  
 $\leftarrow |00\rangle, |01\rangle, |10\rangle, |11\rangle$  (resp.)

Output state of the first two qubits in the four cases:

Case of  $f_{00}$ ?  $|\psi_{00}\rangle = -|00\rangle + |01\rangle + |10\rangle + |11\rangle$

Case of  $f_{01}$ ?  $|\psi_{01}\rangle = +|00\rangle - |01\rangle + |10\rangle + |11\rangle$

Case of  $f_{10}$ ?  $|\psi_{10}\rangle = +|00\rangle + |01\rangle - |10\rangle + |11\rangle$

Case of  $f_{11}$ ?  $|\psi_{11}\rangle = +|00\rangle + |01\rangle + |10\rangle - |11\rangle$

What noteworthy property do these states have? **Orthogonal!**

**Challenge Exercise:** simulate the above  $U$  in terms of  $H$ , CNOT and NOT gates

# one-out-of- $N$ search?

**Natural question:** what about search problems in spaces larger than **four** (and without uniqueness conditions)?

For spaces of size **eight** (say), the previous method breaks down—the state vectors will not be orthogonal

Later on, we'll see how to search a space of size  $N$  with  $O(\sqrt{N})$  queries ...



# Constant vs. balanced

# 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  $\sum_x f(x) = 2^{n-1}$

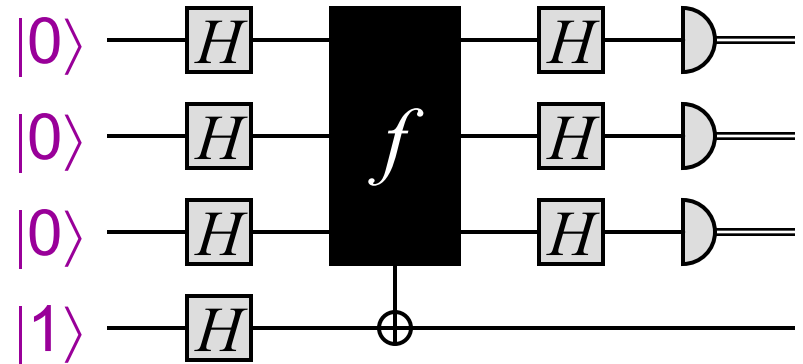
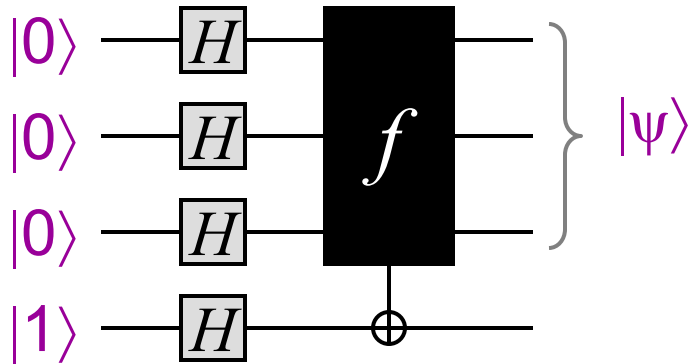
**Goal:** determine whether  $f$  is constant or balanced

How many queries are there needed **classically?** \_\_\_\_\_

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

**Quantumly?** \_\_\_\_\_

# 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:  $H^{\otimes n}|\psi\rangle = \pm |00\dots 0\rangle$

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

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  $\frac{1}{2}$

By repeating the above procedure  $k$  times:

$2k$  queries and one-sided error probability  $(\frac{1}{2})^k$

Therefore, for large  $n$ ,  $\ll 2^n$  queries are likely sufficient

**Introduction to  
Quantum Information Processing  
QIC 710 / CS 768 / PH 767 / CO 681 / AM 871**

**Lecture 5 (2019)**

**Richard Cleve**

QNC 3129

[cleve@uwaterloo.ca](mailto:cleve@uwaterloo.ca)

$H \otimes H \otimes \dots \otimes H$

# About $H \otimes H \otimes \dots \otimes H = H^{\otimes n}$

**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  $x \cdot y = x_1 y_1 \oplus \dots \oplus x_n y_n$

**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  $x \in \{0, 1\}$ ,  $H|x\rangle = |0\rangle + (-1)^x |1\rangle = \sum_{y=0}^1 (-1)^{xy} |y\rangle$

Thus,  $H^{\otimes n} |x_1 \dots x_n\rangle = \left( \sum_{y_1=0}^1 (-1)^{x_1 y_1} |y_1\rangle \right) \dots \left( \sum_{y_n=0}^1 (-1)^{x_n y_n} |y_n\rangle \right)$   
 $= \sum_{y_1 \dots y_n} (-1)^{x_1 y_1 \oplus \dots \oplus x_n y_n} |y_1 \dots y_n\rangle$  ■

# Viewing $\{0,1\}^n$ as a vector space

$\mathbb{Z}_2 = \{0, 1\}$  with addition and multiplication mod 2 is a **field**

$\{0, 1\}^n = \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$  is a  $n$ -dimensional vector space over  $\mathbb{Z}_2$

The usual vector space notions like subspace and dimension make sense here

$x \cdot y = x_1y_1 \oplus \cdots \oplus x_ny_n$  is like a “dot product” of the vectors

$x \cdot y = 0$  can be interpreted as the vectors being “**orthogonal**”

but this notion of orthogonality has some weird properties,

such as the possibility that  $x \cdot x = 0$  even for non-zero vectors

**Caution:** for  $n$ -qubit systems, do not conflate the  $n$ -dimensional vector space  $\{0, 1\}^n$  with the  $2^n$ -dimensional Hilbert space



# Simon's problem

# Quantum vs. classical separations

black-box problem	quantum	classical
constant vs. balanced	<b>1</b> (query)	<b>2</b> (queries)
1-out-of-4 search	<b>1</b>	<b>3</b>
constant vs. balanced	<b>1</b>	$\frac{1}{2} 2^n + 1$
<b>Simon's problem</b>		

(only for exact)

(probabilistic)

# Simon's problem

Let  $f: \{0,1\}^n \rightarrow \{0,1\}^n$  have the property that there exists an  $r \in \{0,1\}^n$  such that  $f(x) = f(y)$  iff  $x \oplus y = r$  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  $r$  in this case?** \_\_\_\_\_

**Answer:**  $r = 101$

# A classical algorithm for Simon

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

1. Choose  $x_1, x_2, \dots, x_k \in \{0,1\}^n$  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  $r$  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  $r$

How big does  $k$  have to be for the probability of a collision to be a constant, such as  $3/4$ ?

**Answer:** order  $2^{n/2}$  (each  $(x_i, x_j)$  collides with prob.  $O(2^{-n})$ )

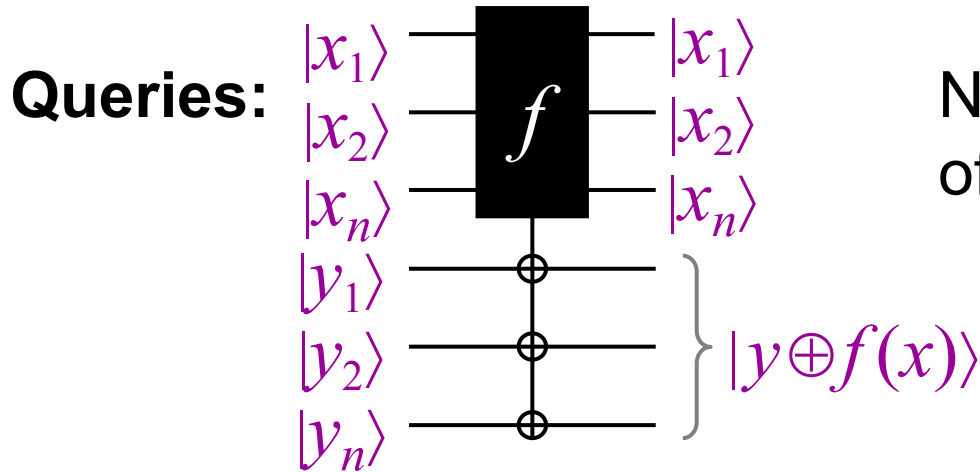
# Classical lower bound

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

Proof is omitted here—note that 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?

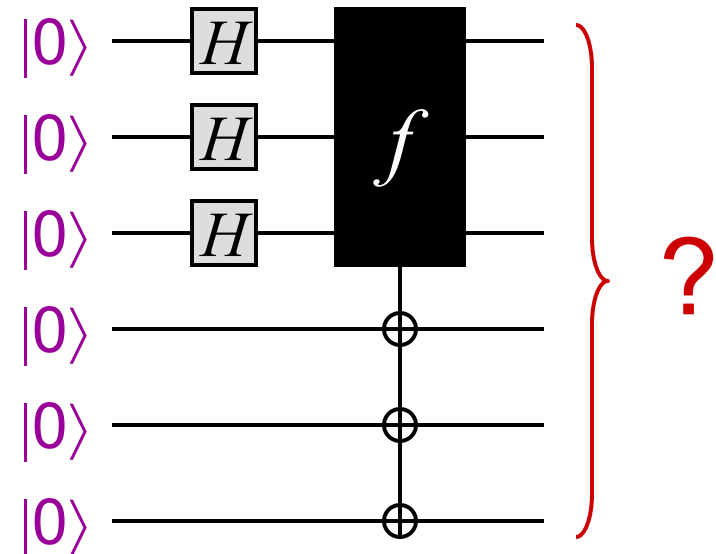
# A quantum algorithm for Simon I



Not clear what *eigenvector* of target registers is ...

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 II

**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  $r \neq 00\dots 0$ )

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

Then the output state can be written as:

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

$$\begin{aligned} & \sum_{x \in T} |x\rangle |f(x)\rangle + |x \oplus r\rangle |f(x \oplus r)\rangle \\ &= \sum_{x \in T} (|x\rangle + |x \oplus r\rangle) |f(x)\rangle \end{aligned}$$

# A quantum algorithm for Simon III

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

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

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

$$\begin{aligned} & \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle + \sum_{y \in \{0,1\}^n} (-1)^{(x \oplus r) \cdot y} |y\rangle \\ &= \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} \left( 1 + (-1)^{r \cdot y} \right) |y\rangle \end{aligned}$$

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



# A quantum algorithm for Simon IV

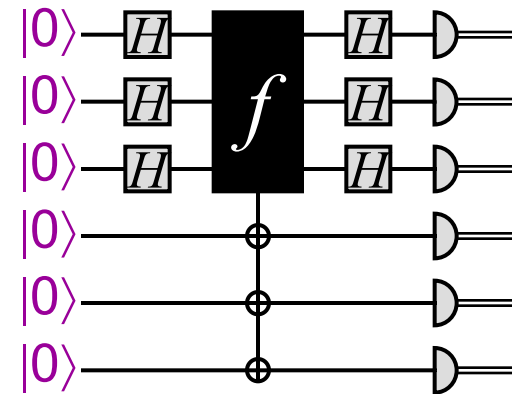
Executing this algorithm  $k = O(n)$  times yields random  $y_1, y_2, \dots, y_k \in \{0,1\}^n$  such that  $r \cdot y_1 = r \cdot y_2 = \dots = r \cdot y_n = 0$

How does this help?

This is a system of  $k$  linear equations:

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{kn} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

With high probability, there is a unique non-zero solution that is  $r$  (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  $\frac{3}{4}$
- 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  $\frac{3}{4}$

black-box problem	quantum	classical
constant vs. balanced	<b>1</b> (query)	<b>2</b> (queries)
1-out-of-4 search	<b>1</b>	<b>3</b>
constant vs. balanced	<b>1</b>	<b>4</b>
<b>Simon's problem</b>	<b><math>O(n^3)</math></b>	<b><math>\Omega(2^{n/2})</math></b>

} for success probability  $\geq \frac{3}{4}$