Bernstein-Vazirani cont.
Simon's Problem

PHYS/CSCI 3090

Prof. Alexandra Kolla

Alexandra.Kolla@Colorado.edu
ECES 122

Prof. Graeme Smith

Graeme.Smith@Colorado.edu
JILA S326

https://home.cs.colorado.edu/~alko5368/indexCSCI3090.html

Come see us!

- Alexandra Kolla/ Graeme Smith: Friday 3:00-4:00 pm, JILA X317.
- Ariel Shlosberg: Tu/Th 2:00-4:00pm, DUANG2B90 (physics help room)
- Steven Kordonowy: Th 11am-12pm, ECAE 124.

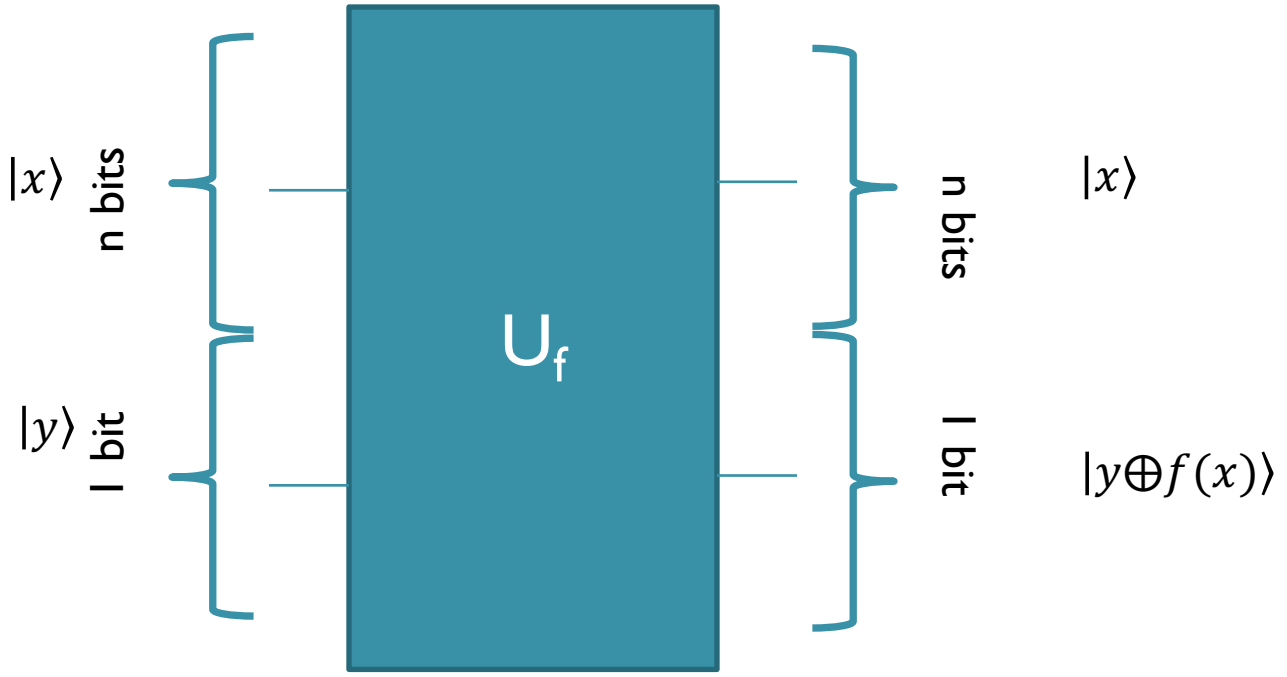
Last Class

- Bernstein-Vazirani
- Another "Quantum supremacy" result

Today

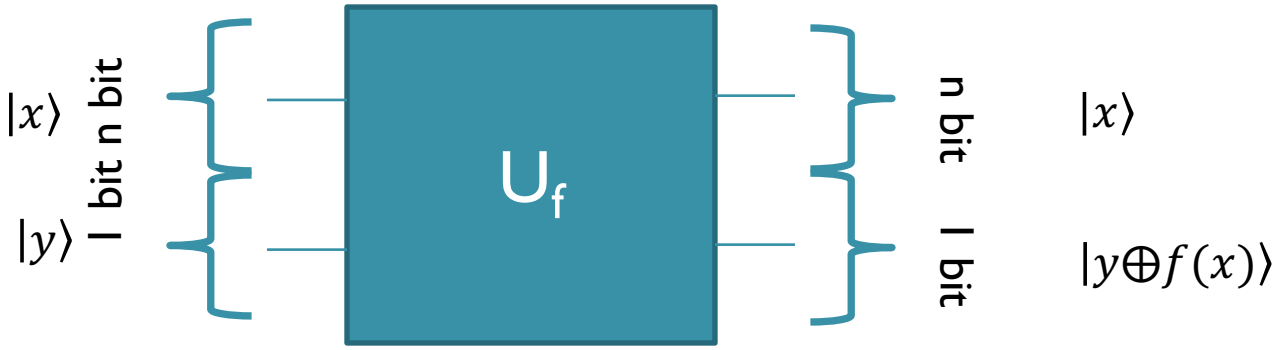
- Finish Bernstein Vazirani
- Simon's problem
- While Bernstein Vazirani gets linear speedup on quantum computer, we can achieve exponential speedup for Simon's problem

The setup, in quantum



U_f applied to the computational basis state $|x\rangle_n|y\rangle_1$ flips the value y of the output register iff f(x)=1.

The Trick



- $U_f |x\rangle_n \otimes 1/\sqrt{2}\ (|0\rangle - |1\rangle) =$
  $(-1)^{f(x)}\ |x\rangle_n \otimes 1/\sqrt{2}\ (|0\rangle - |1\rangle)$

So taking the 1-qubit output register to be $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, we convert a bit flip to a sign change!

The Second Trick

$$H^{\otimes n} |x\rangle_n = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{y \cdot x} |y\rangle_n$$

Because (-1) is raised to the power $\sum x_i y_i$, all that matters is its value mod 2.

Putting everything together (the algorithm)

1. Prepare the input and output registers:

2. Apply the function oracle:

Putting everything together (the algorithm)

1. Prepare the input and output registers:

$$(H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n |1\rangle_1 \quad = \left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\right) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

2. Apply the function oracle:

$$U_f (H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n |1\rangle_1 \quad =$$

$$U_f \left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\right) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =$$

$$\left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} (-1)^{f(x)} |x\rangle_n\right) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Putting everything together (the algorithm)

1. Prepare the input and output registers:
2. Apply the function oracle:

$$U_f(H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n |1\rangle_1 \quad =$$

$$\left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} (-1)^{f(x)} |x\rangle_n\right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

3. Apply Hadamard to the input register:

$$(H^{\otimes n} \otimes 1)\mathrm{Uf}(H^{\otimes n} \otimes \mathrm{H})|0\rangle_n |1\rangle_1 =$$

Putting everything together (the algorithm)

1. Prepare the input and output registers:
2. Apply the function oracle: $U_f(H^{\otimes n} \otimes \mathrm{H})$
   $|0\rangle_n|1\rangle_1 = (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} (-1)^{f(x)}|x\rangle_n)$
   $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
3. Apply Hadamard to the input register:

$$(H^{\otimes n} \otimes 1)\mathrm{Uf}(H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n|1\rangle_1 =$$
$$(\frac{1}{2^{n/2}} H^{\otimes n} \sum_{0<x\leq 2^n} (-1)^{f(x)}|x\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$= (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n, 0<y\leq 2^n} (-1)^{f(x)+x\cdot y}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Putting everything together (the algorithm)

1. Prepare the input and output registers:
2. Apply the function oracle: $U_f(H^{\otimes n} \otimes \mathrm{H})$

   $|0\rangle_n|1\rangle_1 = (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} (-1)^{f(x)}|x\rangle_n)$

   $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
3. Apply Hadamard to the input register:

$(H^{\otimes n} \otimes 1)\mathrm{Uf}(H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n|1\rangle_1 =$

$= (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n, 0<y\leq 2^n} (-1)^{f(x)+x\cdot y}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

$= (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n, 0<y\leq 2^n} (-1)^{a\cdot x+x\cdot y}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

$= (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n, 0<y\leq 2^n} (-1)^{x\cdot (y+a)}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Math stuff

Let a =(a1,a2), y= (y1,y2) be arbitrary 2-bit strings such that y is not the same as x. Let $f(x) = a \cdot x$. What is $\sum_{x_1,x_2=0}^{1}(-1)^{(a+y)\cdot x}$?

A)0

B)4

C) -4

D) $a \cdot y$

More Math stuff

$$\sum_{x=0}^{2^n-1}(-1)^{(a+y)\cdot x} = \prod_{j=1}^{n}\sum_{x_j=0}^{1}(-1)^{(a_j+y_j)x_j}$$

If a and y are different, then the sum vanishes!!

Meaning, the final state of the algorithm is:

$$\sum_y \sum_x (-1)^{x\cdot(y+a)}|y\rangle_n \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =$$

$$|a\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Measure input register, learn a in one query!

Today

- Finish Bernstein Vazirani
- Simon's problem
- While Bernstein Vazirani gets linear speedup on quantum computer, we can achieve exponential speedup for Simon's problem

Two-to-one functions

Let $f: \{0,1\}^n \to \{0,1\}^m$ be a two-to-one function. What is m?

A) n

B) $n-1$

C) 1

D) 2

Two-to-one functions

Simon's problem is concerned with a function $f: \{0,1\}^n \rightarrow \{0,1\}^{n-1}$ that is two-to-one, as follows:

$f(x) = f(y)$ if and only if the n-bit integers x and y are related by $x = y \oplus a$, or, equivalently, $x \oplus y = a$

Simon's problem

- One is told that f is periodic under bitwise modulo-2 addition, $\mathrm{f}(\mathrm{x} \oplus a) = f(x), for\ all\ x$
- The problem is to find the period a.
- Precursor to Shor's factoring, where we are interested in functions that are periodic under ordinary addition (decimal).

Simon's problem

- Classically?
- Ask different x_i until we stumble upon two x_i, x_j that give the same value of f.

Elimination

Let $f: \{0,1\}^n \to \{0,1\}^{n-1}$ be our function as before, and a the hidden "period". Assume I have asked my oracle to give me $f(\mathrm{x})$ for m different values of x, $x_1, \ldots, x_m$. How many values of a have I ruled out at this stage?

A) $\frac{1}{2}m(m-1)$

B) m^2

C) m

D) 1

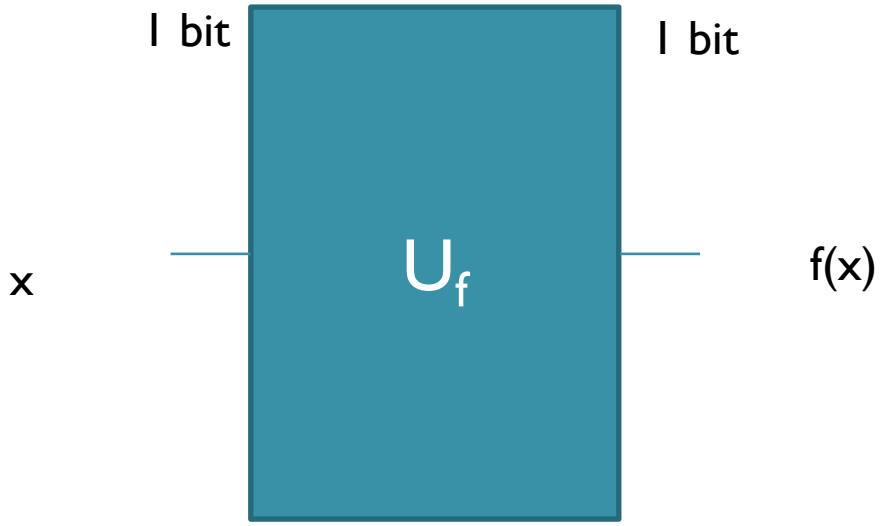
Simon's problem

- Classically?
- Ask different x_i until we stumble upon two x_i,x_j that give the same value of f.
- After asking for m different values of x, I have eliminated at most $\frac{1}{2}m(m-1)$ values for a, since $a \neq x_i \oplus x_j$ for any pair of those values.
- There are total $2^n - 1$ possibilities for a, so I am unlikely to succeed until m becomes of the order of $2^{\frac{n}{2}}$.
- So the number of times I need to run the subroutine grows exponentially with n.

Simon's problem

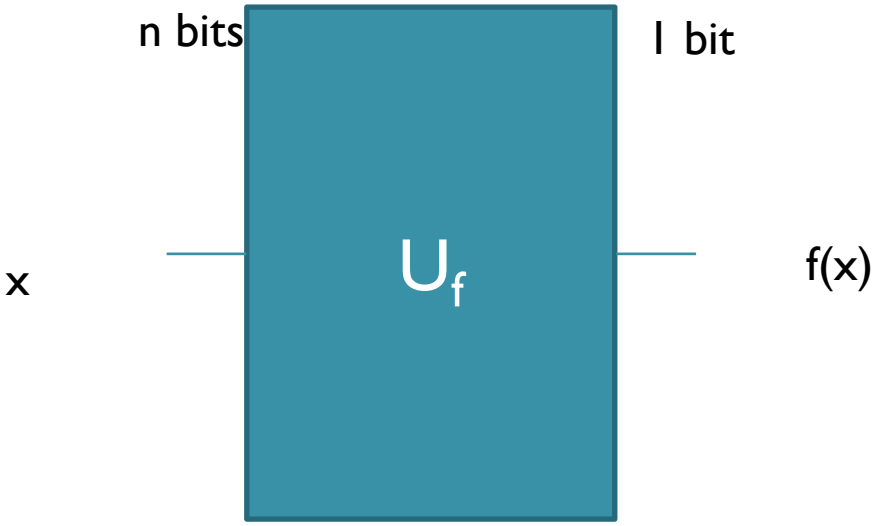
- Quantumly?
- We will see we can determine a with very high probability, only with a linear number of times (not much more than n times)

The setup last week

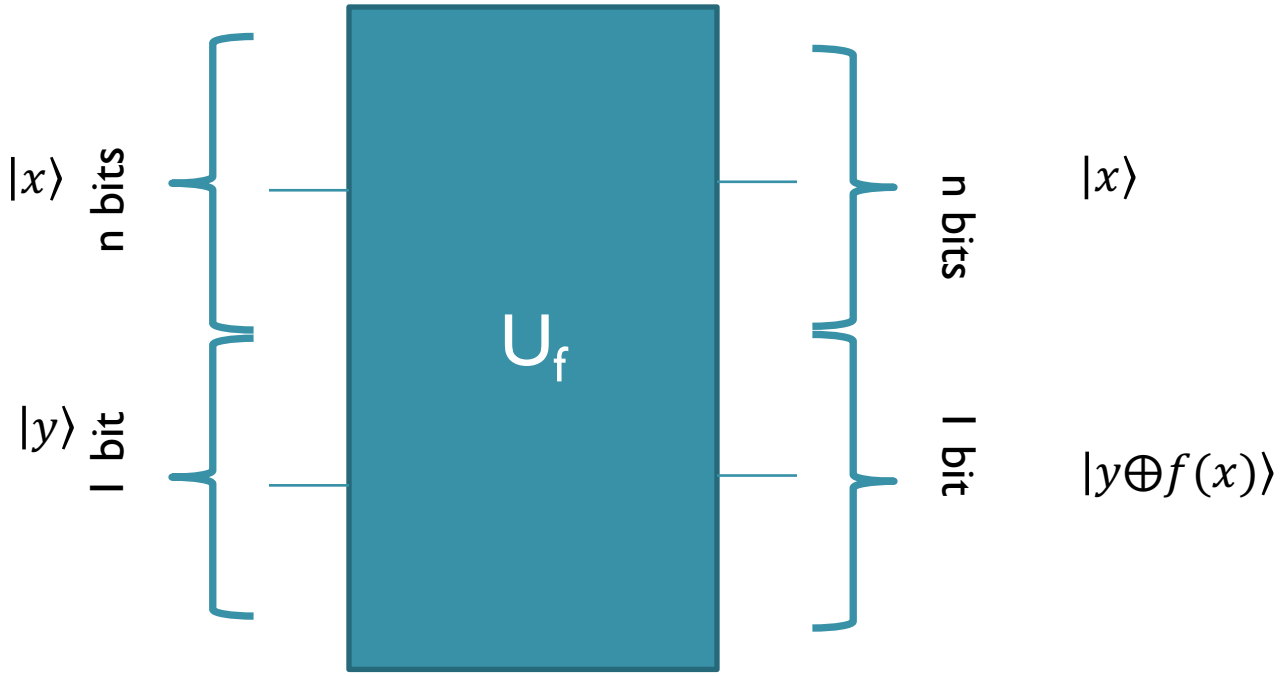


- Both input and output registers contain one bit.
- Functions f that take one bit to one bit
- Two different ways to think about such f.

The setup now

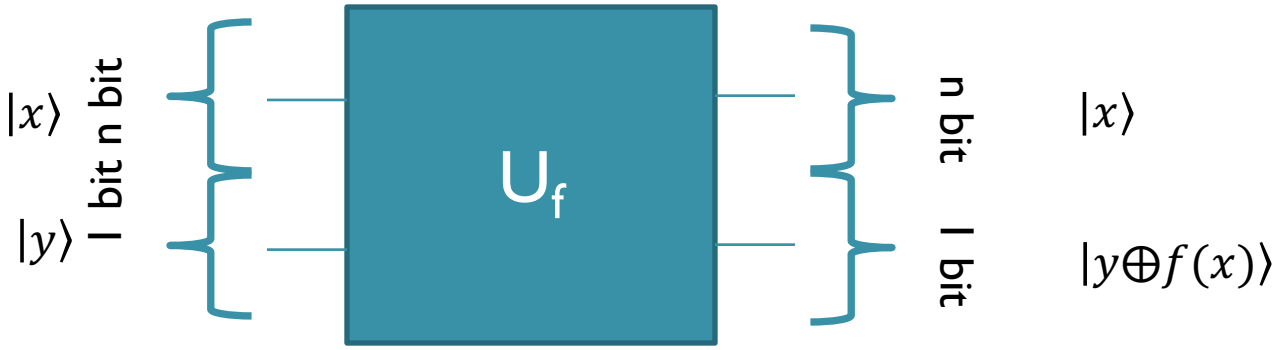


The setup, in quantum



U_f applied to the computational basis state $|x\rangle_n|y\rangle_1$ flips the value y of the output register iff f(x)=1.

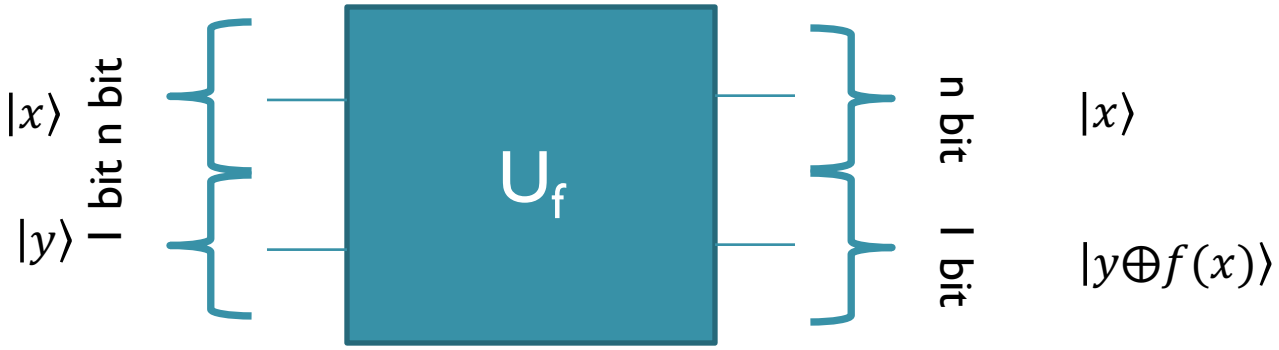
The Trick



- $U_f|x\rangle_n \otimes |0\rangle = |x\rangle_n \otimes |0 \oplus f(x)\rangle =$
  $-|x\rangle_n \otimes |0\rangle$, if f(x) =0
  $-|x\rangle_n \otimes |1\rangle$, if f(x) =1

- $U_f|x\rangle_n \otimes |1\rangle = |x\rangle_n \otimes |1 \oplus f(x)\rangle =$
  $-|x\rangle_n \otimes |1\rangle$, if f(x) =0
  $-|x\rangle_n \otimes |0\rangle$, if f(x) =1

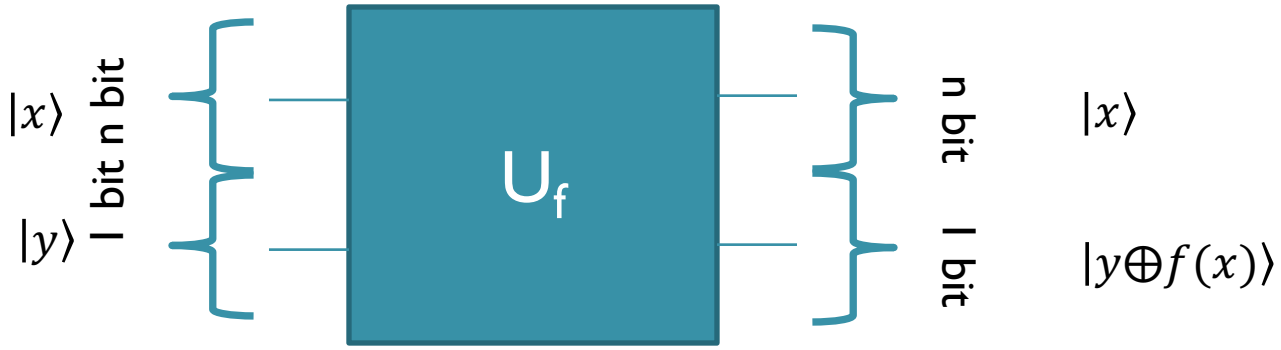
The Trick



- $U_f|x\rangle_n \otimes (|0\rangle + |1\rangle) = ??$

- $U_f|x\rangle_n \otimes (|0\rangle - |1\rangle) = ??$

The Trick



- $U_f |x\rangle_n \otimes 1/\sqrt{2}\ (|0\rangle - |1\rangle) =$
  $(-1)^{f(x)}\ |x\rangle_n \otimes 1/\sqrt{2}\ (|0\rangle - |1\rangle)$

So taking the 1-qubit output register to be $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, we convert a bit flip to a sign change!

The Second Trick

- Recall: $H^{\otimes n} \ |0\rangle_n \quad = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n$

Hadamard

- What is $\mathrm{H}|x\rangle_1$, where x is either in the $|0\rangle$ or $|1\rangle$ state?

A) $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

B) $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

C) $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle)$

D) $|x\rangle$

The Second Trick

- Recall: $H^{\otimes n}\ |0\rangle_n \quad = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n$
- By previous slide,

$$\mathrm{H}|x\rangle_1 = \frac{1}{\sqrt{2}} \sum_{y=0,1} (-1)^{xy} |y\rangle$$

- Generalizing to n qubits:

The Second Trick

$$\mathrm{H}|x\rangle_1 = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle) =$$

$$\frac{1}{\sqrt{2}}\sum_{y=0,1}(-1)^{xy}|y\rangle$$

- Generalizing to 2 qubits:

$$H^{\otimes 2}\ |x\rangle_2 \quad =$$

Hadamard, II

- What is $H^{\otimes n}\ |x\rangle_n$, where x is in one of the 2^n basis states of n qubits?

A) $\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n}|x\rangle_n$

B) $-\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n}|x\rangle_n$

C) $\frac{1}{2^{n/2}}\sum_{y=0}^{2^n-1}(-1)^{y\cdot x}\ |y\rangle_n$

D) $|x\rangle_n$

The Second Trick

- Recall: $H^{\otimes n}\ |0\rangle_n \quad = \frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n$
- By previous slide,

$$\mathrm{H}|x\rangle_1 = \frac{1}{\sqrt{2}} \sum_{y=0,1} (-1)^{xy} |y\rangle$$

- Generalizing to n qubits:

$$H^{\otimes n}\ |x\rangle_n \quad = \ \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{y\cdot x}\ |y\rangle_n$$

Because (-1) is raised to the power $\sum x_i y_i$, all that matters is its value mod 2.

Putting everything together (the algorithm)

1. Prepare the input and output registers:

2. Apply the function oracle:

Putting everything together (the algorithm)

1. Prepare the input and output registers:

$$(H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n |1\rangle_1 \quad = \left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\right) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

2. Apply the function oracle:

$$U_f (H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n |1\rangle_1 \quad =$$

$$U_f \left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} |x\rangle_n\right) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =$$

$$\left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} (-1)^{f(x)} |x\rangle_n\right) \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Putting everything together (the algorithm)

1. Prepare the input and output registers:
2. Apply the function oracle:

$$U_f(H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n |1\rangle_1 \quad =$$

$$\left(\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} (-1)^{f(x)} |x\rangle_n\right) \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

3. Apply Hadamard to the input register:

$$(H^{\otimes n} \otimes 1)\mathrm{U_f}(H^{\otimes n} \otimes \mathrm{H})|0\rangle_n |1\rangle_1 =$$

Putting everything together (the algorithm)

1. Prepare the input and output registers:
2. Apply the function oracle: $U_f(H^{\otimes n} \otimes \mathrm{H})$
   $|0\rangle_n|1\rangle_1 = (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n} (-1)^{f(x)}|x\rangle_n)$
   $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
3. Apply Hadamard to the input register:

$$(H^{\otimes n} \otimes 1)\mathrm{U_f}(H^{\otimes n} \otimes \mathrm{H})\ |0\rangle_n|1\rangle_1 =$$
$$(\frac{1}{2^{n/2}} H^{\otimes n} \sum_{0<x\leq 2^n} (-1)^{f(x)}|x\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$= (\frac{1}{2^{n/2}} \sum_{0<x\leq 2^n, 0<y\leq 2^n} (-1)^{f(x)+x\cdot y}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Putting everything together (the algorithm)

1. Prepare the input and output registers:
2. Apply the function oracle: $U_f(H^{\otimes n}\otimes \mathrm{H})$

   $|0\rangle_n|1\rangle_1 = (\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n}(-1)^{f(x)}|x\rangle_n)$

   $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
3. Apply Hadamard to the input register:

$(H^{\otimes n}\otimes 1)\mathrm{U_f}(H^{\otimes n}\otimes \mathrm{H})\ |0\rangle_n|1\rangle_1 =$

$= (\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n,0<y\leq 2^n}(-1)^{f(x)+x\cdot y}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

$= (\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n,0<y\leq 2^n}(-1)^{a\cdot x+x\cdot y}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

$= (\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n,0<y\leq 2^n}(-1)^{x\cdot (y+a)}|y\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Math stuff

Let a =(a1,a2), y= (y1,y2) be arbitrary 2-bit strings such that y is not the same as x. Let $f(x) = a \cdot x$. What is $\sum_{x_1,x_2=0}^{1}(-1)^{(a+y)\cdot x}$?

A)0

B)4

C) -4

D) $a \cdot y$

More Math stuff

$$\sum_{x=0}^{2^n-1}(-1)^{(a+y)\cdot x} = \prod_{j=1}^{n}\sum_{x_j=0}^{1}(-1)^{(a_j+y_j)x_j}$$

If a and y are different, then the sum vanishes!!

Meaning, the final state of the algorithm is:

$$\sum_y \sum_x (-1)^{x\cdot(y+a)}|y\rangle_n \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =$$

$$|a\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Measure input register, learn a in one query!