Bernstein-Vazirani 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

Homework

- HW 3 will be out soon, due next Monday at noon.
- Typo on HW2, we will re-ask this problem on HW3.

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

- Deutsch's problem
- Simplest example of quantum tradeoff that sacrifices particular information to get relational information.
- First "Quantum supremacy" result

Today

- Bernstein-Vazirani
- Another “Quantum supremacy” result

Bitwise Inner Product

Let $x = (x_0, \ldots, x_n)$ and $a = (a_0, \ldots, a_n)$ be two integers, represented as n-bit strings. The bitwise inner product of x and a, denoted $x \cdot a$ modulo 2 is

$$x_0 a_0 \oplus x_1 a_1 \oplus \ldots \oplus x_n a_n$$

Binary arithmetic test

Let $a = a_n \dots a_0$ be an n-bit binary string. What is the number a expressed in the decimal system?

A) $a_n \cdot 2^0 + a_{n-1} \cdot 2^1 + \cdots + a_0 \cdot 2^n$

B) $a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_0 \cdot 2^0$

C) $I\ don't\ know$

D) $a_n \cdot 10^0 + a_{n-1} \cdot 10^1 + \cdots + a_0 \cdot 10^n$

Intergers and Binary Representations

- What is the m-th bit of a?

A) $a \cdot 2^m$

B) $a \oplus 2^m$

C) $a \cdot m$

D)

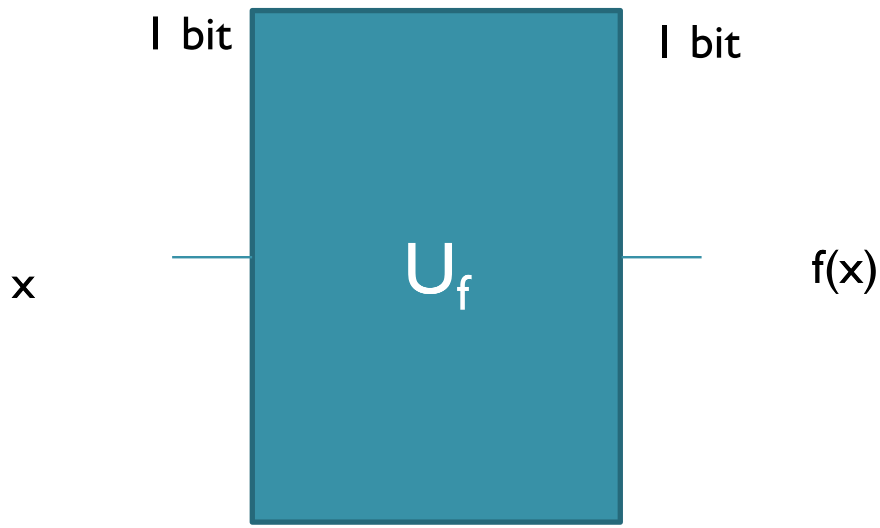
Bernstein-Vazirani

- Let a be an unknown non-negative integer less than 2^n.
- Represent it as an n-bit string
- Let $f(x) = a \cdot x$= $x_0 a_0 \oplus x_1 a_1 \oplus \ldots \oplus x_n a_n$
- Suppose we have an oracle (subroutine) that when you ask x, it gives you f(x).
- How many times do we need to call the oracle to determine a?

Bernstein-Vazirani

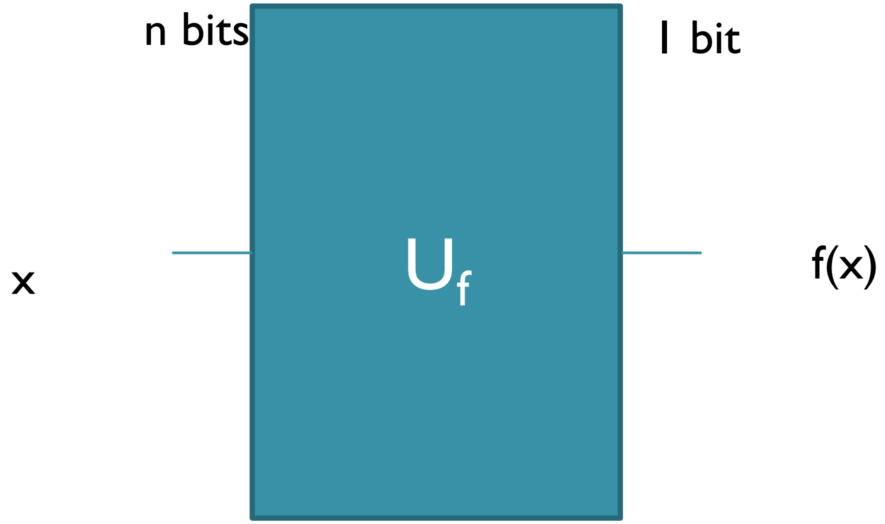
- Classically?
- We could learn the n bits of a by applying f to the n values $x = 2^m, 0 \leq m < n$.
- n invocations of the subroutine!
- With quantum we can ask once.

The setup last week

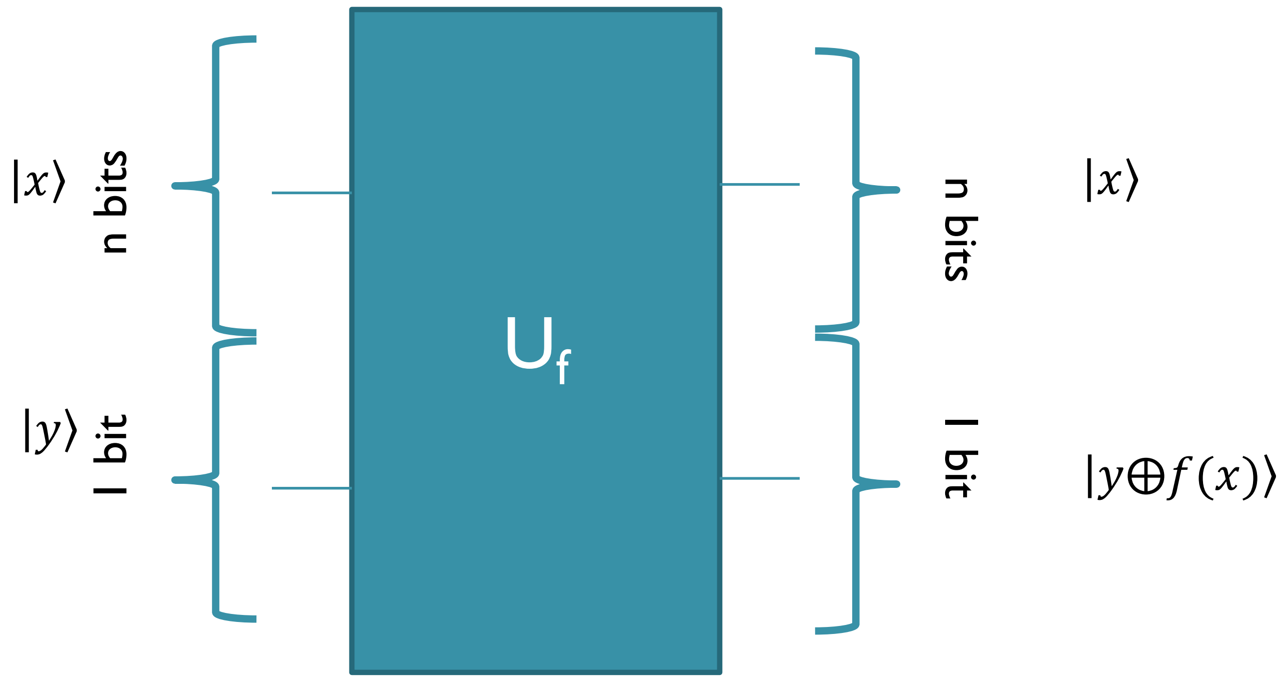


- Both input and output registers contain one bit.
- Functions f that take one bit to one bit
- Two different ways to think about such f.

The setup now

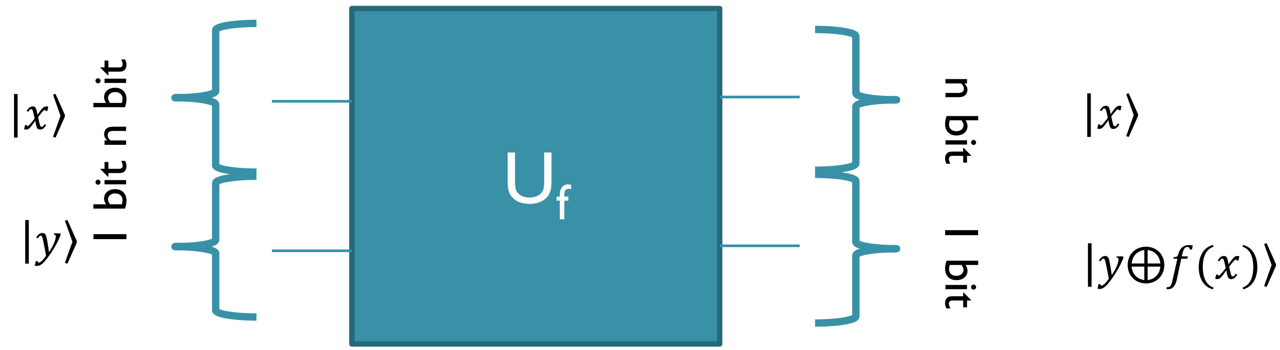


The setup, in quantum



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.

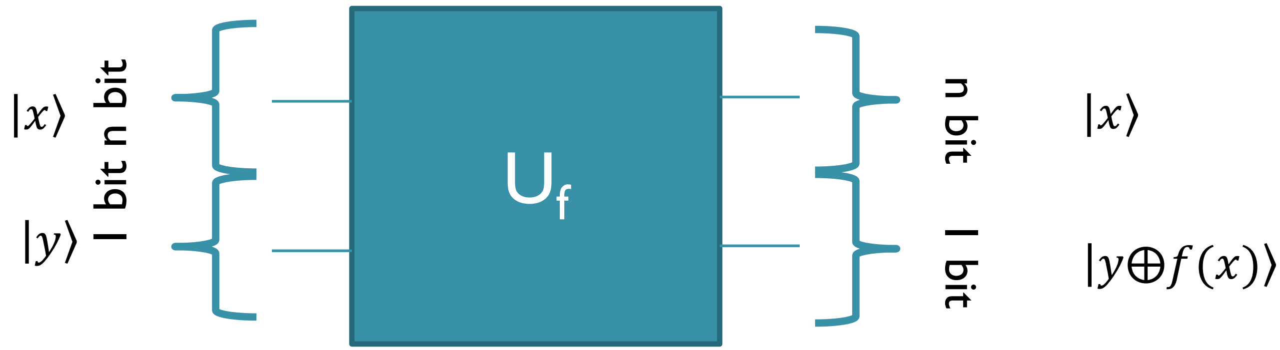
The Trick



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

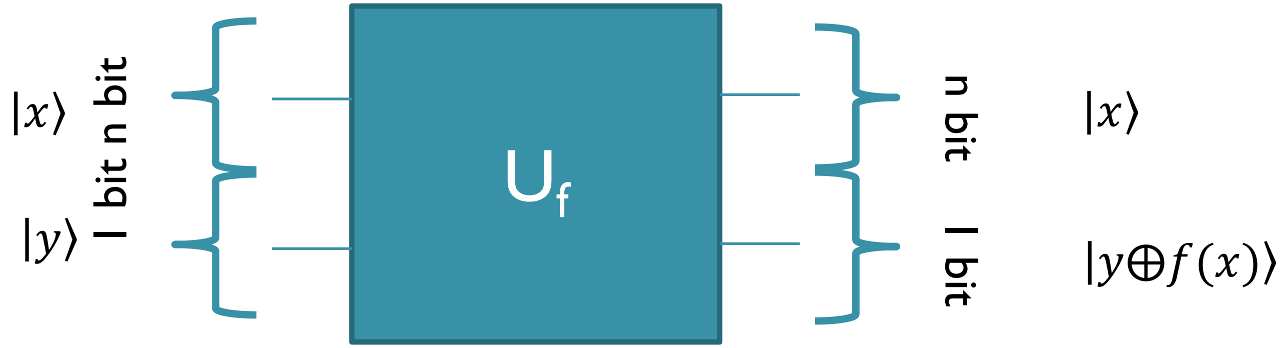
The Trick



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

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

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:
$(H^{\otimes n}\otimes \mathrm{H})\ |0\rangle_n|1\rangle_1 \quad = (\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n}|x\rangle_n)$ $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
2. Apply the function oracle:
$$U_f(H^{\otimes n}\otimes \mathrm{H})\ |0\rangle_n|1\rangle_1 \quad =$$
$U_f(\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n}|x\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =$
$(\frac{1}{2^{n/2}}\sum_{0<x\leq 2^n}(-1)^{f(x)}|x\rangle_n)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Putting everything together (the algorithm)

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

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

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

3. Apply Hadamard to the input register:

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

Putting everything together (the algorithm)

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