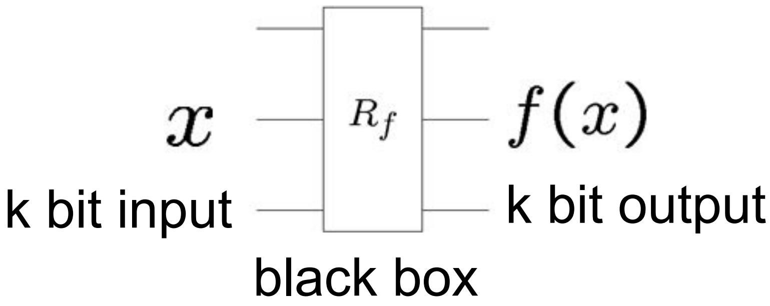
Practical Quantum Computing

Lecture 6
The early algorithms: Bernstein - Vazirani

with slides from Dave Bacon https://homes.cs.washington.edu/~dabacon/teaching/siena/

Classical Promise Problem Query Complexity

Given: A black box which computes some function



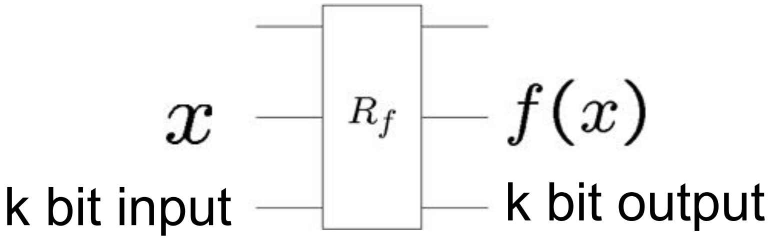
Promise: the function belongs to a set $\mathcal{S}$ which is a subset of all possible functions.

Properties: the set $\mathcal{S}$ can be divided into disjoint subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m$

Problem: What is the minimal number of times we have to use (query) the black box in order to determine which subset $\mathcal{S}_i$ the function belongs to?

Functions

We can write the unitary



in outer product form as

$$U_f = \sum_{x=0}^{2^n-1} |f(x)\rangle\langle x|$$

so that

$$\begin{aligned} U_f|y\rangle &= \left(\sum_{x=0}^{2^n-1} |f(x)\rangle\langle x|\right)|y\rangle \\ &= \sum_{x=0}^{2^n-1} |f(x)\rangle\langle x|y\rangle = \sum_{x=0}^{2^n-1} |f(x)\rangle\delta_{y,x} = |f(y)\rangle \end{aligned}$$

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$

Functions

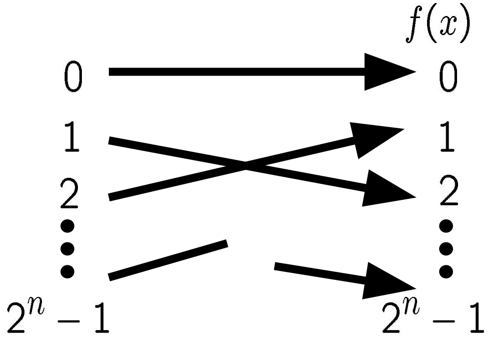
Note that the transform is unitary

$$U_f^\dagger = \left(\sum_{x=0}^{2^n-1} |f(x)\rangle\langle x|\right)^\dagger = \sum_{x=0}^{2^n-1} (|f(x)\rangle\langle x|)^\dagger = \sum_{x=0}^{2^n-1} |x\rangle\langle f(x)|$$

$$U_f U_f^\dagger = \left(\sum_{x=0}^{2^n-1} |f(x)\rangle\langle x|\right)\left(\sum_{y=0}^{2^n-1} |y\rangle\langle f(y)|\right)$$

$$= \sum_{x,y=0}^{2^n-1} |f(x)\rangle\langle x|y\rangle\langle f(y) = \sum_{x,y=0}^{2^n} |f(x)\rangle\langle f(y)|\delta_{x,y}$$

$$= \sum_{x=0}^{2^n-1} |f(x)\rangle\langle f(x)| = I$$



precisely when f(x) is one to one!

Quantum Algorithms

David Deutsch

Richard Jozsa

1992: Deutsch-Jozsa Algorithm

Exact classical query complexity: $2^{n-1}+1$

Bounded error classical query complexity: $O(1)$

Exact quantum q. complexity: 1

Umesh Vazirani

Ethan Bernstein

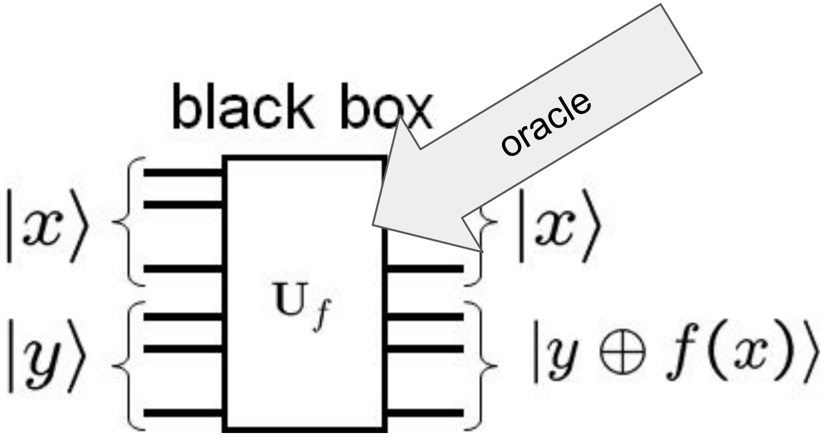
1993: Bernstein-Vazirani Algorithm (non-recursive)

Exact classical query complexity: n

Bounded error classical query complexity: $\Omega(n)$

Exact quantum q. complexity: 1

Query Complexity



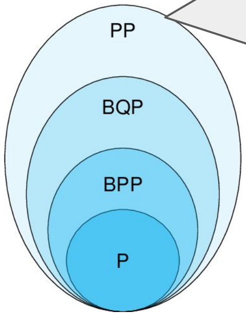
$$f : x \in \{0,1\}^n \to \{0,1\}^k$$

	probability	
Exact classical query complexity	0	Bounded error algorithms are allowed to fail with a bounded probability of failure.
Bounded error classical query complexity	1/3	
Exact quantum query complexity	0	
Bounded error quantum query complexity	1/3	

BPP, BQP

Informally, a problem is in **BPP** (bounded-error probabilistic polynomial time) if there is an algorithm for it that has the following properties:

- is allowed to flip coins and make random decisions
- is guaranteed to run in polynomial time
- on any given run of the alg probability of at most 1/3 c wrong answer, whethe or NO.



In complexity theory, PP is the class of decision problems solvable by a probabilistic Turing machine in polynomial time, with an error probability of less than 1/2 for all instances. The abbreviation PP refers to probabilistic polynomial time. A PP algorithm is permitted to have a probability that depends on the input size, whereas BPP does not.

Informally, a decision problem is a member of **BQP** (bounded-error quantum polynomial time) if there exists a quantum algorithm (an algorithm that runs on a quantum computer):

- that solves the decision problem with high probability
- is guaranteed to run in polynomial time
- a run of the algorithm will correctly solve the decision problem with a probability of at least 2/3.

It is the quantum analogue to the complexity class BPP

Bernstein-Vazirani Problem

Given: A function with n bit strings as input and one bit as output

$$f : x \in \{0,1\}^n \rightarrow \{0,1\}$$

Promise: The function is of the form

$$f(x) = (a \cdot x) \oplus b \qquad a \in \{0,1\}^n \qquad b \in \{0,1\}$$

$$y \cdot x = y_1 x_1 \oplus y_2 x_2 \oplus \cdots \oplus y_n x_n$$

Problem: Find the n bit string a

Bernstein-Vazirani Problem

Given: A function with n bit strings as input and one bit as output

$$f : x \in \{0, 1\}^n \rightarrow \{0, 1\}$$

Promise: The function is of the form

$$f(x) = (a \cdot x) \oplus b \qquad a \in \{0, 1\}^n \qquad b \in \{0, 1\}$$

$$y \cdot x = y_1 x_1 \oplus y_2 x_2 \oplus \cdots \oplus y_n x_n$$

Problem: Find the n bit string a

Notice that the querying f yields a single bit of information. But we need n bits of information to describe a.

Classical Bernstein-Vazirani

Notice that the querying f yields a single bit of information. But we need n bits of information to describe a.

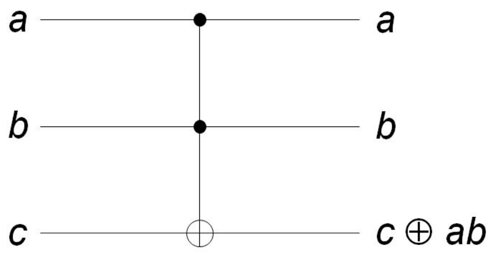
Classically, the most efficient method to find the secret string is by evaluating the function n times with the input values $x = 2^i$ for all $i \in \{0, 1, \ldots, n-1\}$

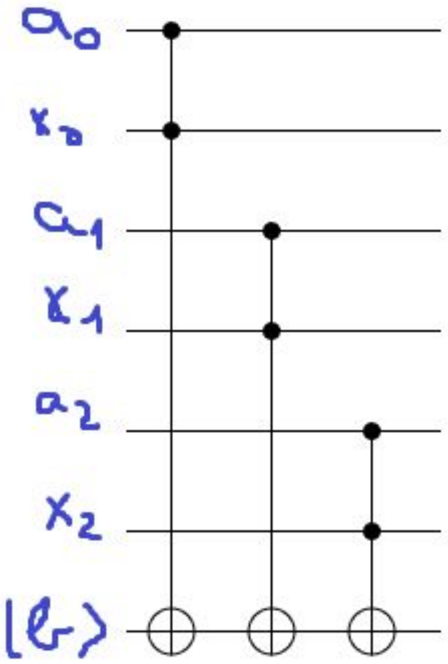
$$
\begin{aligned}
f(1000\cdots0_n) &= s_1 \\
f(0100\cdots0_n) &= s_2 \\
f(0010\cdots0_n) &= s_3 \\
&\vdots \\
f(0000\cdots1_n) &= s_n
\end{aligned}
$$

Implement the oracle

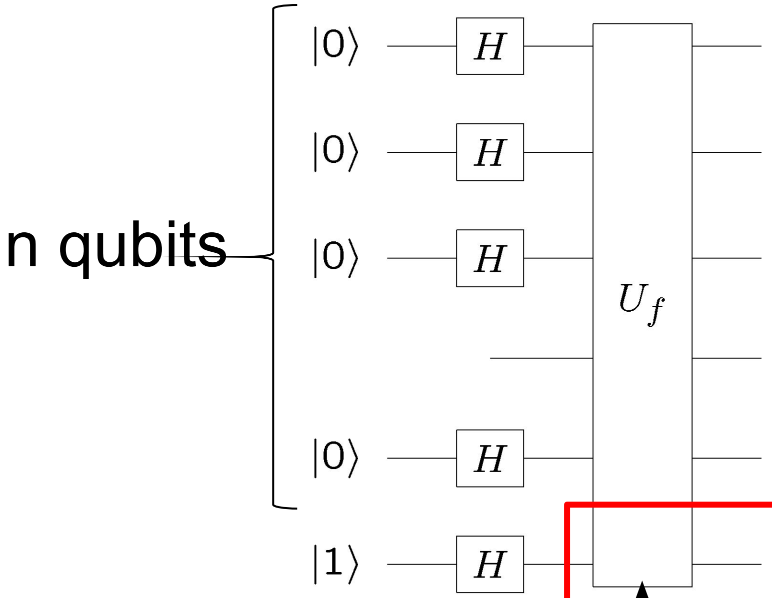
$$f(x) = (a \cdot x) \oplus b$$

$$y \cdot x = y_1 x_1 \oplus y_2 x_2 \oplus \cdots \oplus y_n x_n$$





Quantum Bernstein-Vazirani



$$\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{f(y)}|y\rangle = \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+b}|y\rangle$$

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

for phase kickback

$$H^{\otimes n}|0\rangle \otimes H|1\rangle = \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}|y\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Quantum Bernstein-Vazirani

Show the phase kickback

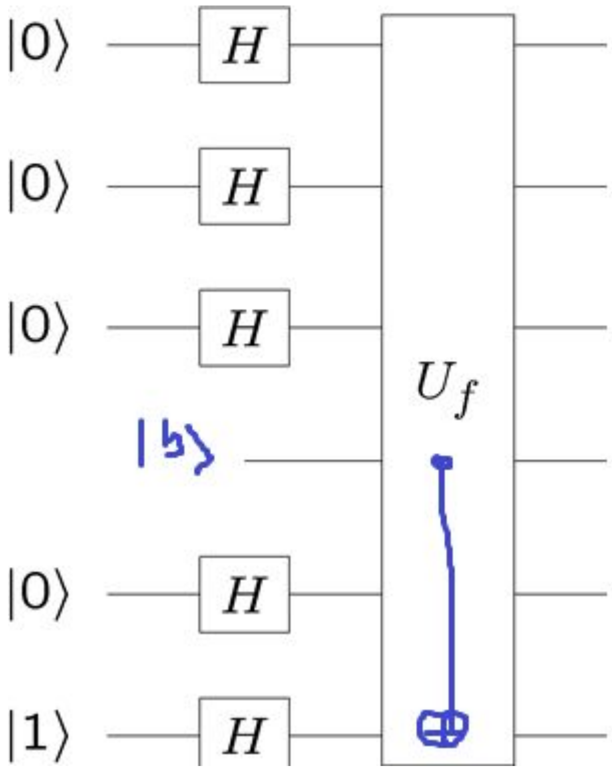
|Register>|b>|->

if |b> == |0> (when f(x) == 0)

+|Register>|b>**|->**

elif |b> == |1> (when f(x) == 1)

-|Register>|b>**|->**



Hadamard it! (Interference)

$$H^{\otimes n}\left[\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+b}|y\rangle\right] = \frac{1}{2^n}\sum_{x,y=0}^{2^n-1}(-1)^{a\cdot y+b}(-1)^{x\cdot y}|x\rangle$$

$$|0\rangle \xrightarrow{H} |0\rangle + |1\rangle$$
$$|1\rangle \xrightarrow{H} |0\rangle - |1\rangle \quad \Big\}\ |0\rangle + (-1)^{xb}|b\rangle$$

$$|x\rangle \xrightarrow{H^{\otimes |x|}} \sum_{0}^{2^{|x|}-1}(-1)^{xy}|y\rangle$$

$$|1\rangle \longrightarrow (-1)^{1\cdot 0}|0\rangle + (-1)^{1\cdot 1}|1\rangle$$
$$= |0\rangle - |1\rangle \qquad \sum(-1)^{xb}|b\rangle$$

$$= \frac{(-1)^b}{2^n}\sum_{x=0}^{2^n-1}\left(\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+x\cdot y}\right)|x\rangle$$

Hadamard it! (Interference)

$$H^{\otimes n}\left[\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+b}|y\rangle\right] = \frac{1}{2^n}\sum_{x,y=0}^{2^n-1}(-1)^{a\cdot y+b}(-1)^{x\cdot y}|x\rangle$$

$$= \frac{(-1)^b}{2^n}\sum_{x=0}^{2^n-1}\left(\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+x\cdot y}\right)|x\rangle$$

$$\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+x\cdot y} = \sum_{y_1,\ldots,y_n=0}^{1}(-1)^{a_1y_1+\cdots+a_ny_n+x_1y_1+\cdots+x_ny_n}$$

$$= \sum_{y_1=0,1}\cdots\sum_{y_n=0,1}(-1)^{a_1y_1+\cdots+a_ny_n+x_1y_1+\cdots+x_ny_n}$$

$$= ((-1)^0+(-1)^{a_1+x_1})\sum_{y_2=0,1}\cdots\sum_{y_n=0,1}(-1)^{a_2y_2+\cdots+a_ny_n+x_2y_2+\cdots+x_ny_n}$$

$$= 2^n\delta_{a_1,x_1}\delta_{a_2,x_2}\cdots\delta_{a_n,x_n} = 2^n\delta_{a,x}$$

Hadamard it! (Interference)

$$H^{\otimes n}\left[\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+b}|y\rangle\right] = \frac{1}{2^n}\sum_{x,y=0}^{2^n-1}(-1)^{a\cdot y+b}(-1)^{x\cdot y}|x\rangle$$

$$= \frac{(-1)^b}{2^n}\sum_{x=0}^{2^n-1}\left(\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+x\cdot y}\right)|x\rangle$$

$$\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+x\cdot y} = \sum_{y_1,\ldots,y_n=0}^{1}(-1)^{a_1y_1+\cdots+a_ny_n+x_1y_1+\cdots+x_ny_n}$$

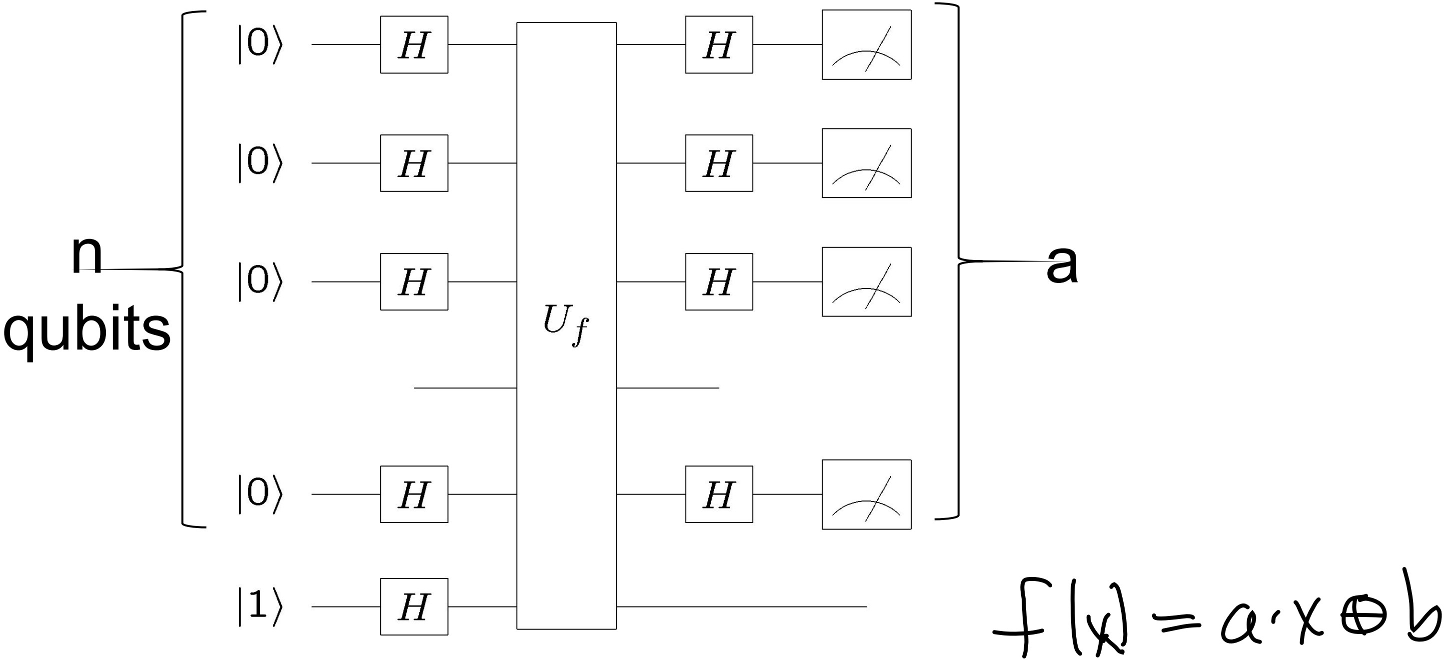
$$= \sum_{y_1=0,1}\cdots\sum_{y_n=0,1}(-1)^{a_1y_1+\cdots+a_ny_n+x_1y_1+\cdots+x_ny_n}$$

$$= \boxed{((-1)^0+(-1)^{a_1+x_1})}\sum_{y_2=0,1}\cdots\sum_{y_n=0,1}(-1)^{a_2y_2+\cdots+a_ny_n+x_2y_2+\cdots+x_ny_n}$$

$$= \boxed{2^n}\delta_{a_1,x_1}\delta_{a_2,x_2}\cdots\delta_{a_n,x_n} = 2^n\delta_{a,x}$$

$$H^{\otimes n}\left[\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{a\cdot y+b}|y\rangle\right] = (-1)^b\sum_{x=0}^{2^n}\delta_{a,x}|x\rangle = (-1)^b|a\rangle$$

Quantum Bernstein-Vazirani



We can determine a using only a single quantum query!