Introduction to
Quantum Information Processing

Richard Cleve

Institute for Quantum Computing

and

Cheriton School of Computer Science

Lecture 8

The discrete log problem

Topics included: the definition of the discrete log problem and an efficient quantum algorithm for it

Preliminaries: $\mathbb{Z}_p$ and $\mathbb{Z}_p^*$

For any prime p, we define two sets: $\mathbb{Z}_p = \{0, 1, 2, \dots, p-1\}$ and $\mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$

It's natural to perform $\bmod p$ arithmetic on these sets:

- $\mathbb{Z}_p$ is a **field** with respect to addition and multiplication modulo p
- $\mathbb{Z}_p^*$ is a **group** with respect to multiplication modulo p

A **generator** of $\mathbb{Z}_p^*$ is $g \in \mathbb{Z}_p^*$ such that $\mathbb{Z}_p^* = \{ g^0, g^1, g^2, \dots, g^{p-2} \}$

the set of exponents is $\{0, 1, 2, \dots, p-2\} = \mathbb{Z}_{p-1}$

and $g^x g^y = g^{x+y \bmod p-1}$

Examples: for $p = 7$, we have $\mathbb{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$

- 2 is ***not*** a generator of $\mathbb{Z}_7^*$ because $\{2^0, 2^1, 2^2, 2^3, 2^4, 2^5\} = \{1, 2, 4, 1, 2, 4\} = \{1, 2, 4\}$
- 3 ***is*** a generator of $\mathbb{Z}_7^*$ because $\{3^0, 3^1, 3^2, 3^3, 3^4, 3^5\} = \{1, 3, 2, 6, 4, 5\}$

Also, for arbitrary m (not necessarily prime), $\mathbb{Z}_m = \{0, 1, 2, \dots, m-1\}$

Discrete log & exp problem

Relative to a prime modulus p, for a generator $g \in \mathbb{Z}_p^*$, we define two functions/problems:

Discrete exponential function

$\exp_g : \mathbb{Z}_{p-1} \to \mathbb{Z}_p^*$ is defined as

$\exp_g(r) = g^r$ (modulo p)

→

Discrete exp problem

input: p (n-bit prime), g (generator of $\mathbb{Z}_p^*$), $r \in \mathbb{Z}_{p-1}$

output: $s = g^r$

Classical gate cost is $O(n^2 \log n)$

Discrete logarithm function

$\log_g : \mathbb{Z}_p^* \to \mathbb{Z}_{p-1}$ is defined as

$\log_g(s) = r$ such that $g^r = s$

→

Discrete log problem (DLP)

input: p (n-bit prime), g (generator of $\mathbb{Z}_p^*$), $s \in \mathbb{Z}_p^*$

output: $r \in \mathbb{Z}_{p-1}$ such that $g^r = s$

No efficient classical algorithm for **DLP** is known (and it's presumed hardness is the basis of cryptosystems)

Shor's quantum algorithm solves this at cost $O(n^2 \log n)$

Discrete log and Simon

DLP input: p (prime), g (generator of $\mathbb{Z}_p^*$), $s \in \mathbb{Z}_p^*$ **output:** $r = \log_g(s)$

Shor's clever idea: define $f: \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1} \to \mathbb{Z}_p^*$ as $f(a_1, a_2) = g^{a_1} s^{-a_2} \bmod p$

When is $f(a_1, a_2) = f(b_1, b_2)$?

Theorem: $f(a_1, a_2) = f(b_1, b_2)$ if and only if $(a_1, a_2) - (b_1, b_2) = k(r, 1)$ for some $k \in \mathbb{Z}_{p-1}$

Simon's property is like the modulo 2 case of this in n dimensions:

$f(a) = f(b)$ if and only if $a \oplus b \in \{0^n, r\}$

$k(r_1, \dots, r_n)$ for $k \in \mathbb{Z}_2$

$(a_1, \dots, a_n) - (b_1, \dots, b_n) \bmod 2$

Proof of the theorem

DLP input: p (prime), g (generator of $\mathbb{Z}_p^*$), $s \in \mathbb{Z}_p^*$ **output:** $r = \log_g(s)$

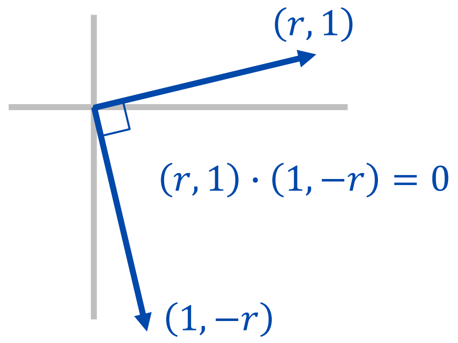
Shor's clever idea: define $f: \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1} \to \mathbb{Z}_p^*$ as $f(a_1, a_2) = g^{a_1} s^{-a_2} \bmod p$

When is $f(a_1, a_2) = f(b_1, b_2)$?

Theorem: $f(a_1, a_2) = f(b_1, b_2)$ if and only if $(a_1, a_2) - (b_1, b_2) = k(r, 1)$ for some $k \in \mathbb{Z}_{p-1}$

Proof:

Since $s = g^r$, we have $s^{a_2} = g^{ra_2}$ and $f(a_1, a_2) = g^{a_1} s^{-a_2} = g^{a_1} g^{-ra_2} = g^{a_1 - ra_2}$



Therefore, $f(a_1, a_2) = f(b_1, b_2)$ iff $a_1 - ra_2 = b_1 - rb_2 \pmod{p-1}$

iff $(a_1, a_2) \cdot (1, -r) = (b_1, b_2) \cdot (1, -r)$

iff $\big((a_1, a_2) - (b_1, b_2)\big) \cdot (1, -r) = 0$ ⟵ "orthogonal" to $(1, -r)$

iff $(a_1, a_2) - (b_1, b_2)$ is a multiple of $(r, 1)$

Simon mod m for $f: (\mathbb{Z}_m)^d \to T$

Definition: a function $f: (\mathbb{Z}_m)^d \to T$ is ***m*-to-1** if, for all $a \in (\mathbb{Z}_m)^d$, the set of points in $(\mathbb{Z}_m)^d$ that f maps to $f(a)$ has size m

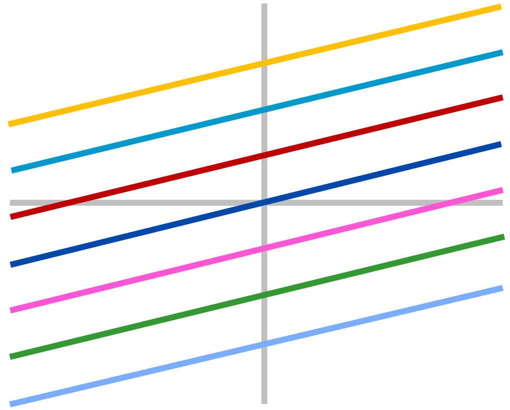
colliding sets: subsets of $(\mathbb{Z}_m)^d$ of size m on which f is constant

Simon mod m property

$f: (\mathbb{Z}_m)^d \to T$ is m-to-1 and there exists $r \in (\mathbb{Z}_m)^d$ for which every colliding set is of the form:

$\{a, a+r, a+2r, \ldots, a+(m-1)r\}$ for some $a \in (\mathbb{Z}_m)^d$

Equivalent to: $f(a) = f(b)$ iff $a - b$ is a multiple of r



schematic for $(\mathbb{Z}_m)^2$

Simon's problem

f is the special case where $m = 2$ and $d = n$

Shor's function in DLP

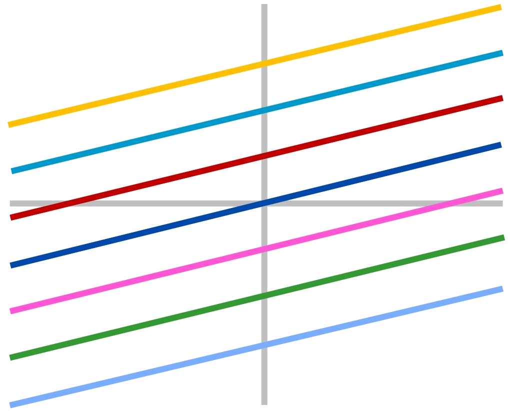
f is special case where $m = p - 1$ and $d = 2$

Simon's problem mod m

Simon mod m property

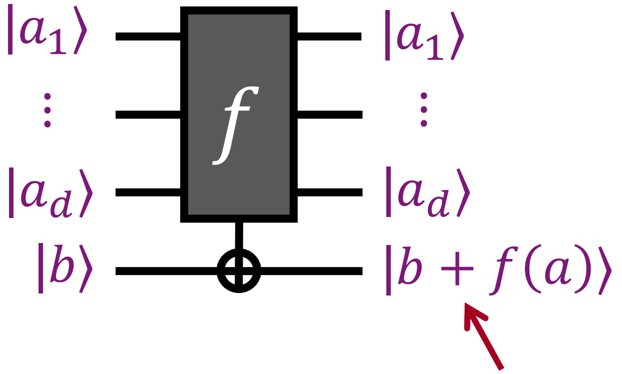
$f: (\mathbb{Z}_m)^d \to T$ is m-to-1 and there exists $r \in (\mathbb{Z}_m)^d$ for which every colliding set is of the form:

$\{a, a+r, a+2r, \dots, a+(m-1)r\}$ for some $a \in (\mathbb{Z}_m)^d$



schematic for $(\mathbb{Z}_m)^2$

Queries:



each register is m-dimensional (not just qubits)

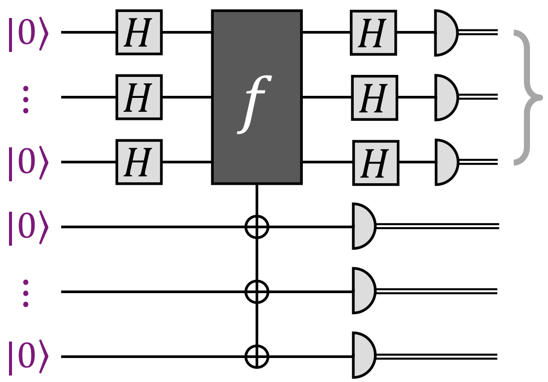
$|T|$-dimensional

give T an additive group structure (e.g. $\mathbb{Z}_{|T|}$)

Goal: to determine r

Simon mod m algorithm (overview)

Recall that Simon's algorithm is based on:



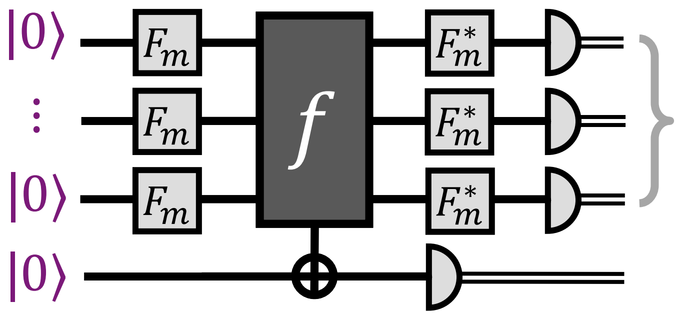
random $b \in \{0,1\}^n$
such that $b \cdot r = 0$

H has a natural m-dimensional analogue:

$$F_m = \frac{1}{\sqrt{m}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{m-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(m-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{m-1} & \omega^{2(m-1)} & \cdots & \omega^{(m-1)^2} \end{bmatrix}$$

where $\omega = e^{2\pi i/m}$ (Fourier transform)

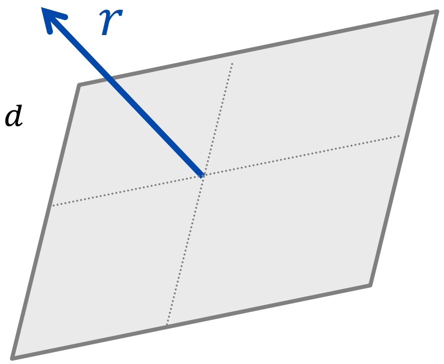
For Simon mod m, we'll try this:



We'll see that the output is similar

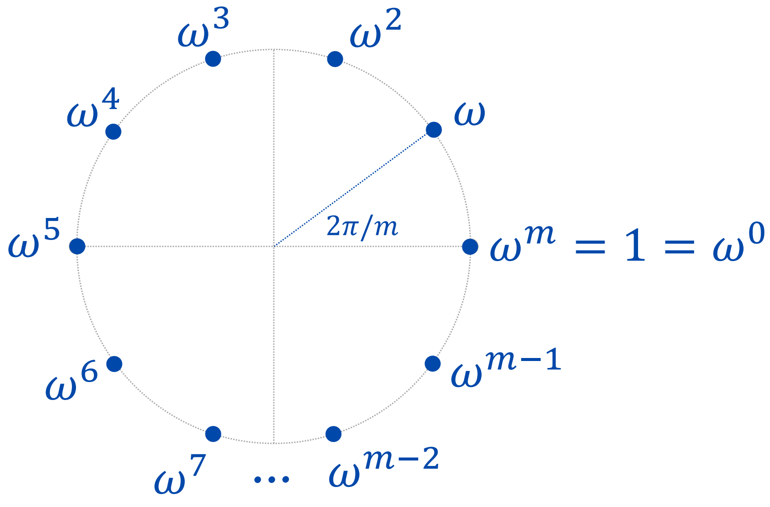
random $b \in (\mathbb{Z}_m)^d$
such that $b \cdot r = 0$

m^{d-1} elements of $(\mathbb{Z}_m)^d$
are "orthogonal" to r



Fourier transform

Primitive m^{th} root of unity: $\omega = e^{2\pi i/m}$



$$\left.\begin{array}{l}
1 + \omega + \omega^2 + \cdots + \omega^{m-1} = 0 \\
1 + \omega^2 + \omega^4 + \cdots + \omega^{2(m-1)} = 0 \\
1 + \omega^3 + \omega^6 + \cdots + \omega^{3(m-1)} = 0 \\
\quad\vdots \\
1 + \omega^{m-1} + \omega^{2(m-1)} + \cdots + \omega^{(m-1)^2} = 0
\end{array}\right\}$$

Exercise: prove these

$$1 + \omega^m + \omega^m + \cdots + \omega^m = m$$

Fourier transform

$$F_m = \frac{1}{\sqrt{m}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{m-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(m-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{m-1} & \omega^{2(m-1)} & \cdots & \omega^{(m-1)^2} \end{bmatrix} \qquad F_m^* = \frac{1}{\sqrt{m}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-m-1} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(m-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(m-1)} & \omega^{-2(m-1)} & \cdots & \omega^{-(m-1)^2} \end{bmatrix}$$

Exercise: prove that F_m is unitary

Fourier transform

$$F_m = \frac{1}{\sqrt{m}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{m-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(m-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{m-1} & \omega^{2(m-1)} & \cdots & \omega^{(m-1)^2} \end{bmatrix}$$

$$F_m^* = \frac{1}{\sqrt{m}}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-m-1} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(m-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(m-1)} & \omega^{-2(m-1)} & \cdots & \omega^{-(m-1)^2} \end{bmatrix}$$

For all $a \in \mathbb{Z}_m$

$$F_m|a\rangle = \frac{1}{\sqrt{m}}\sum_{b\in\mathbb{Z}_m} \omega^{ab}\,|b\rangle$$

$$F_m^*|a\rangle = \frac{1}{\sqrt{m}}\sum_{b\in\mathbb{Z}_m} \omega^{-ab}\,|b\rangle$$

For all $(a_1, a_2) \in \mathbb{Z}_m \times \mathbb{Z}_m$

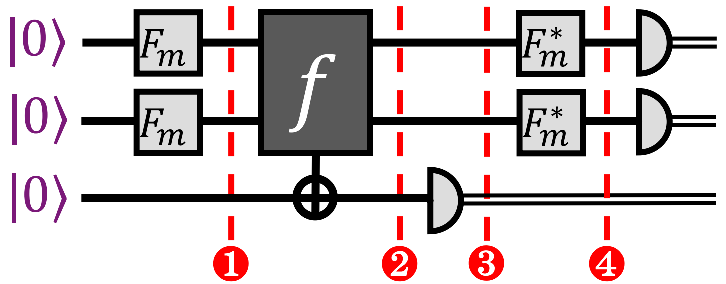
dot product $(a_1, a_2)\cdot(b_1, b_2) = a_1\, b_1 + a_2\, b_2 \bmod m$

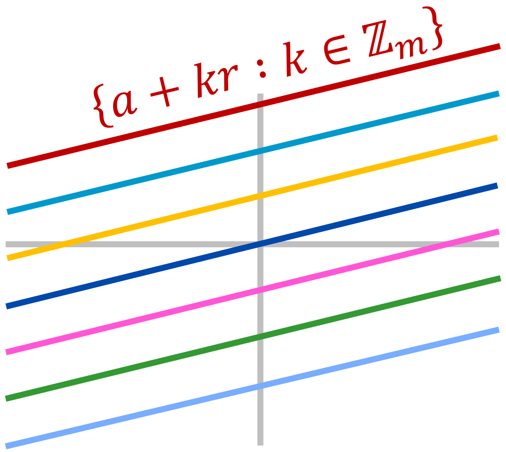
$$F_m \otimes F_m|a_1, a_2\rangle = \sum_{b\in\mathbb{Z}_m^2} \omega^{a\cdot b}|b_1, b_2\rangle$$

$$F_m^* \otimes F_m^*|a_1, a_2\rangle = \sum_{b\in\mathbb{Z}_m^2} \omega^{-a\cdot b}|b_1, b_2\rangle$$

Simon mod m algorithm

Let $f : (\mathbb{Z}_m)^d \rightarrow T$ satisfy the Simon mod m condition (for simplicity, set $d = 2$)





colliding sets of f

❶ $\sum_{(a_1,a_2)\in\mathbb{Z}_m^2} |a_1,a_2\rangle|0\rangle$

❷ $\sum_{(a_1,a_2)\in\mathbb{Z}_m^2} |a_1,a_2\rangle|f(a_1,a_2)\rangle$

❸ uniform superposition over a random colliding set:

$$\sum_{k\in\mathbb{Z}_m} |(a_1,a_2)+k(r_1,r_2)\rangle$$

$$= |a_1,a_2\rangle + |(a_1,a_2)+(r_1,r_2)\rangle + |(a_1,a_2)+2(r_1,r_2)\rangle + \cdots + |(a_1,a_2)+(m-1)(r_1,r_2)\rangle$$

❹ Applying $F_m^* \otimes F_m^*$

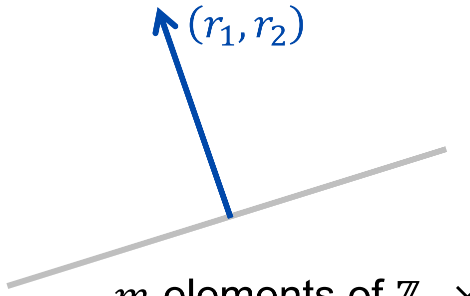
$F_m^* \otimes F_m^*$ applied to the superposition of a random colliding set is:

$$F_m^* \otimes F_m^* \left(\sum_{k \in \mathbb{Z}_m} |(a_1, a_2) + k(r_1, r_2)\rangle \right) = \sum_{k \in \mathbb{Z}_m} F_m^* \otimes F_m^* |(a_1, a_2) + k(r_1, r_2)\rangle$$

$$= \sum_{k \in \mathbb{Z}_m} \left(\sum_{b \in \mathbb{Z}_m^2} \omega^{-(a+kr)\cdot b} |b_1, b_2\rangle \right)$$

$$= \sum_{k \in \mathbb{Z}_m} \left(\sum_{b \in \mathbb{Z}_m^2} \omega^{-a\cdot b} \omega^{-k(r\cdot b)} |b_1, b_2\rangle \right)$$

$$= \sum_{b \in \mathbb{Z}_m^2} \omega^{-a\cdot b} \left(\sum_{k \in \mathbb{Z}_m} \omega^{-k(r\cdot b)} \right) |b_1, b_2\rangle$$

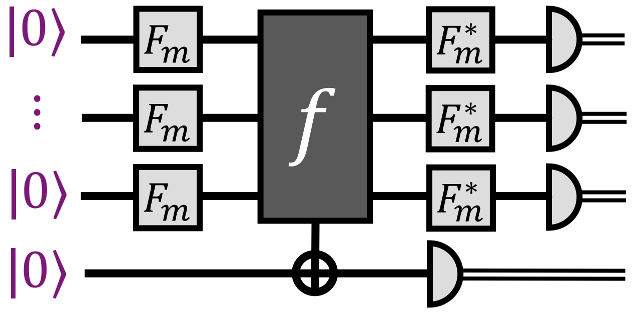


Measuring this state: $\Pr[(b_1, b_2)] = \begin{cases} 0 & \text{if } (b_1, b_2)\cdot(r_1, r_2) \neq 0 \\ 1/m & \text{if } (b_1, b_2)\cdot(r_1, r_2) = 0 \end{cases}$

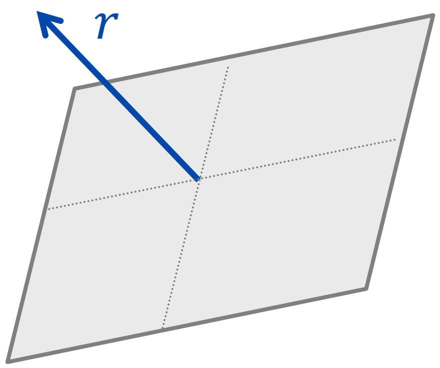
Therefore, the measured result is a random (b_1, b_2) such that $(b_1, b_2)\cdot(r_1, r_2) = 0$

Summary of Simon mod m algorithm

We've shown that if $f: (\mathbb{Z}_m)^d \to T$ has the Simon mod m property then



random $b \in (\mathbb{Z}_m)^d$
such that $b \cdot r = 0$



m^{d-1} elements of $(\mathbb{Z}_m)^d$ that are "orthogonal" to r

From repeated runs of this, there are various ways of determining r

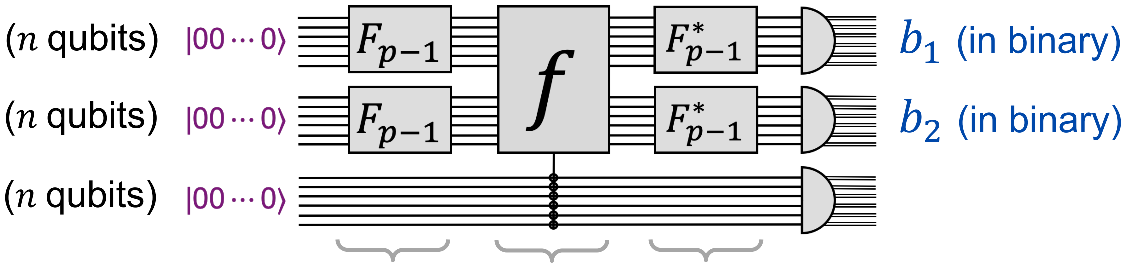
Discrete log ← Simon mod m

Discrete log problem

input: n-bit prime p and $g, s \in \mathbb{Z}_p^*$

output: $r \in \mathbb{Z}_{p-1}$ such that $g^r = s$

We can ***implement*** the query algorithm with qubits and 1- and 2-qubit gates



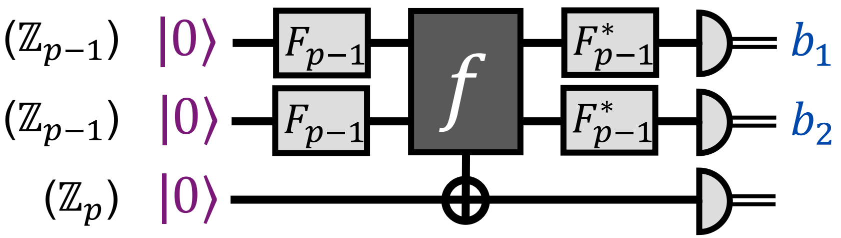
$O(n^2 \log n)$ elementary gates

Back-box problem

input: black-box for $f: \mathbb{Z}_{p-1}^2 \rightarrow \mathbb{Z}_p^*$ such that $f(a_1, a_2) = g^{a_1} s^{-a_2} \bmod p$

output: $r \in \mathbb{Z}_{p-1}$ such that $g^r = s$

Quantum query algorithm



That's the basic idea behind Shor's algorithm for the discrete log problem

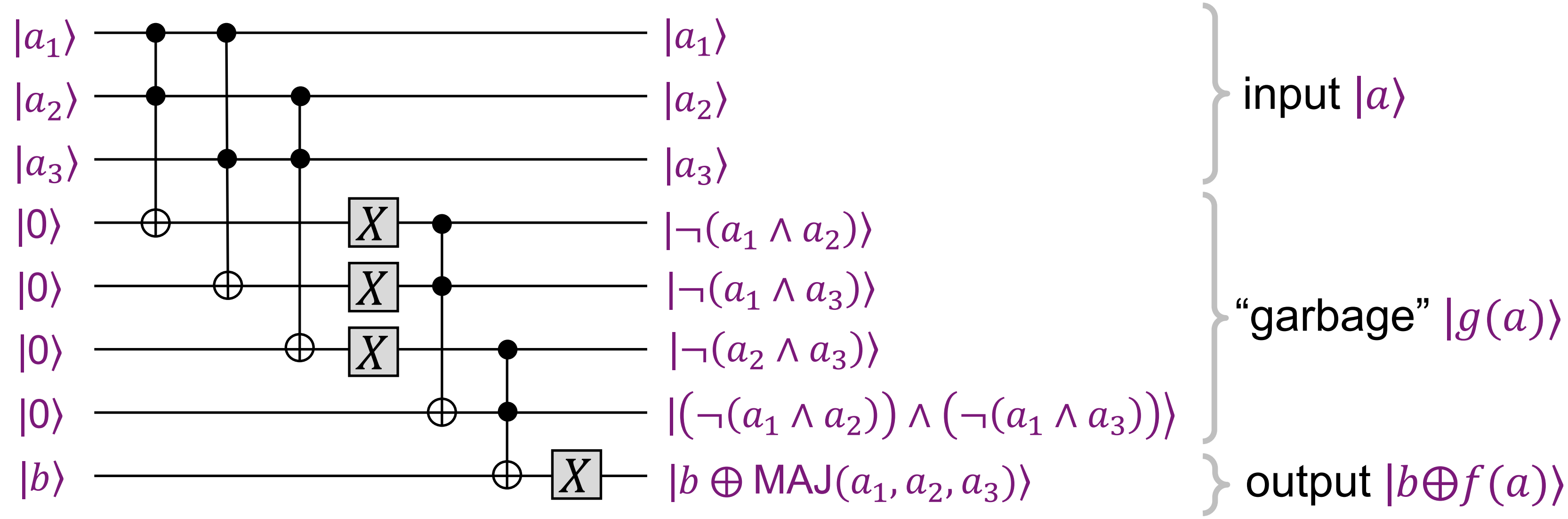
How do we implement the f-query and F_{p-1}?

How **not** to simulate an f-query

If $f:\{0,1\}^{n_1} \to \{0,1\}^{n_2}$ is efficiently computable by a classical circuit, how do we efficiently simulate an f-query $|a\rangle|b\rangle \mapsto |a\rangle|b \oplus f(a)\rangle$ for quantum algorithms?

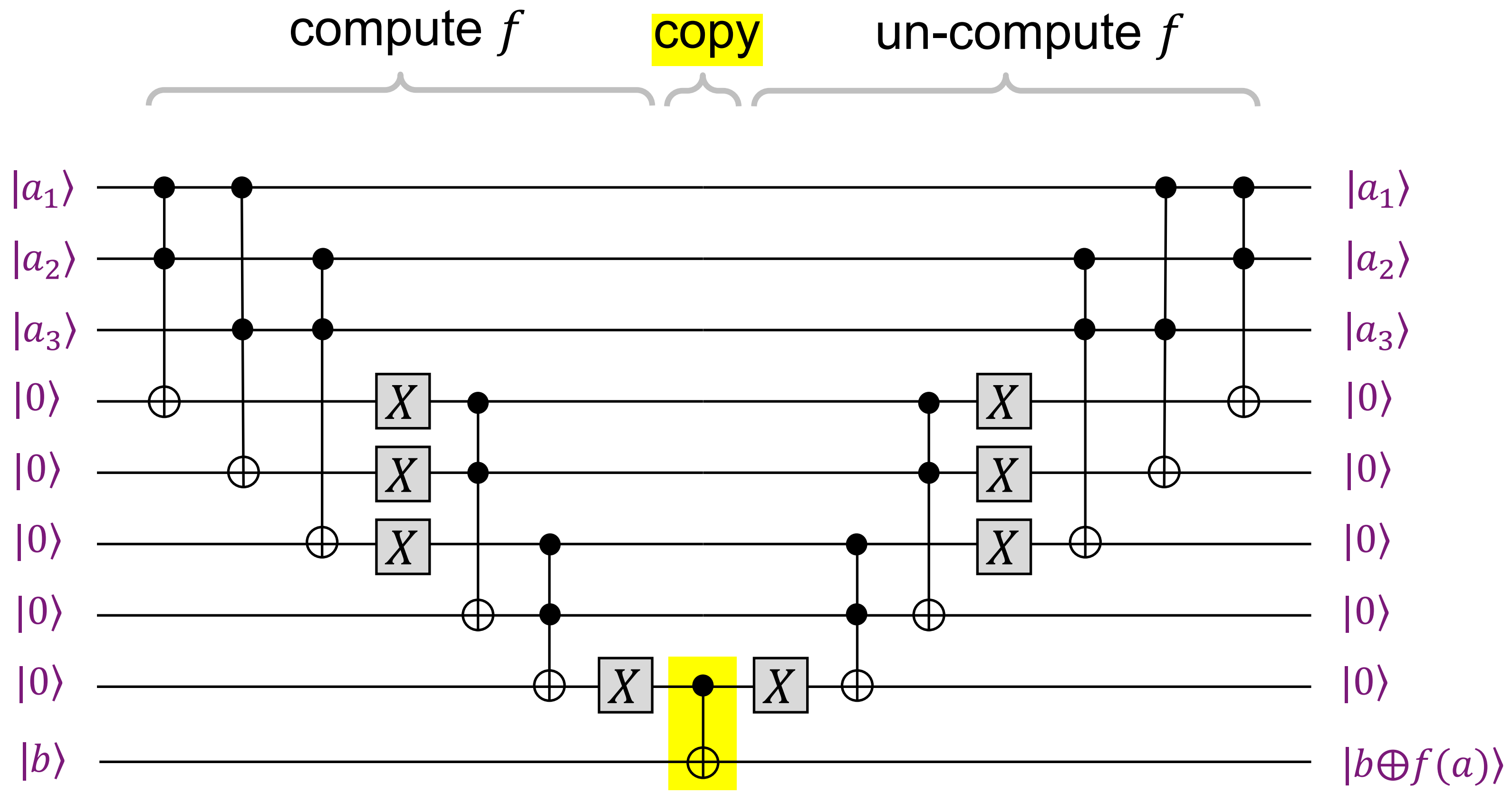
Quantum circuits can simulate classical circuits: $|a\rangle|00\ldots0\rangle|b\rangle \longmapsto |a\rangle|g(a)\rangle|b \oplus f(a)\rangle$
(where the intermediate register is from the Toffoli gates)

Example:



Is this OK for an f-query? No, because $\sum_{a,b} \alpha_{a,b}\, |a\rangle|b \oplus f(a)\rangle|g(a)\rangle \neq \left(\sum_{a,b} \alpha_{a,b}\, |a\rangle|b \oplus f(a)\rangle\right)|g(a)\rangle$

How to simulate an f-query



This is a good f-query because $\sum_{a,b} \alpha_{a,b} \, |a\rangle|b\oplus f(a)\rangle|00\ldots0\rangle = \left(\sum_{a,b} \alpha_{a,b} \, |a\rangle|b\oplus f(a)\rangle\right)|00\ldots0\rangle$

More details of DLP algorithm

Calculating r

We obtain a random (b_1, b_2) such that
$(b_1, b_2) \cdot (r, 1) \equiv 0 \bmod p - 1$

How do we calculate r from (b_1, b_2)?

We can solve for $r = -b_2/b_1 \bmod p - 1$,
if b_1 has an inverse in $\mathbb{Z}_{p-1}$

This is the case if and only if b_1 and $p - 1$ are **relatively prime** ($\gcd(b_1, p-1) = 1$)

The process can be repeated until such a b_1 arises, which occurs with good enough frequency (further details omitted)

Implementing the Fourier transform

Efficiently implementing F_{p-1} is tricky

Instead, Shor implemented F_{2^n}
for the power of 2 nearest to $p - 1$

With careful error-analysis it can be shown that this is good enough in terms of error probability

Next lecture we'll see how to efficiently implement F_{2^n}