

Introduction to Quantum Information Processing

QIC 710 / CS 768 / PH 767 / CO 681 / AM 871

Lecture 9 (2019)

Richard Cleve

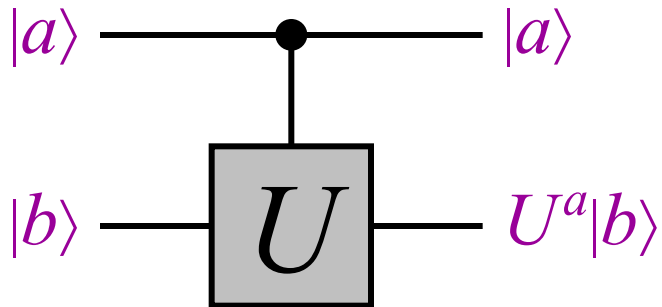
QNC 3129

cleve@cs.uwaterloo.ca

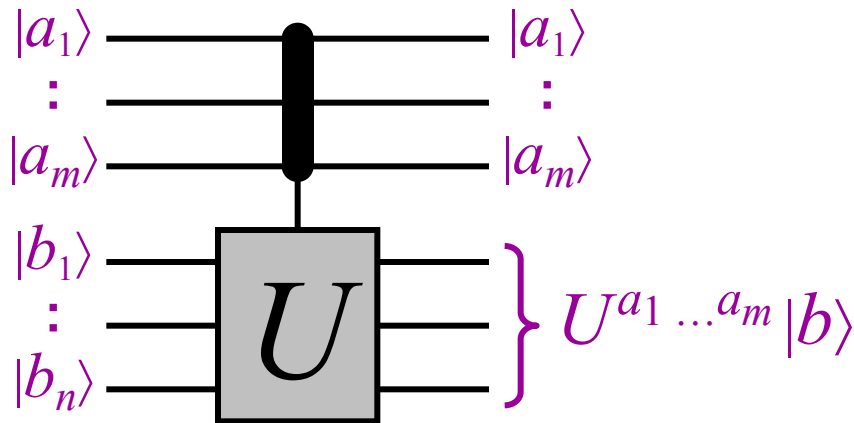
Recap of:

Eigenvalue estimation problem
(a.k.a. phase estimation)

Generalized controlled- U gates



$$\begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix}$$



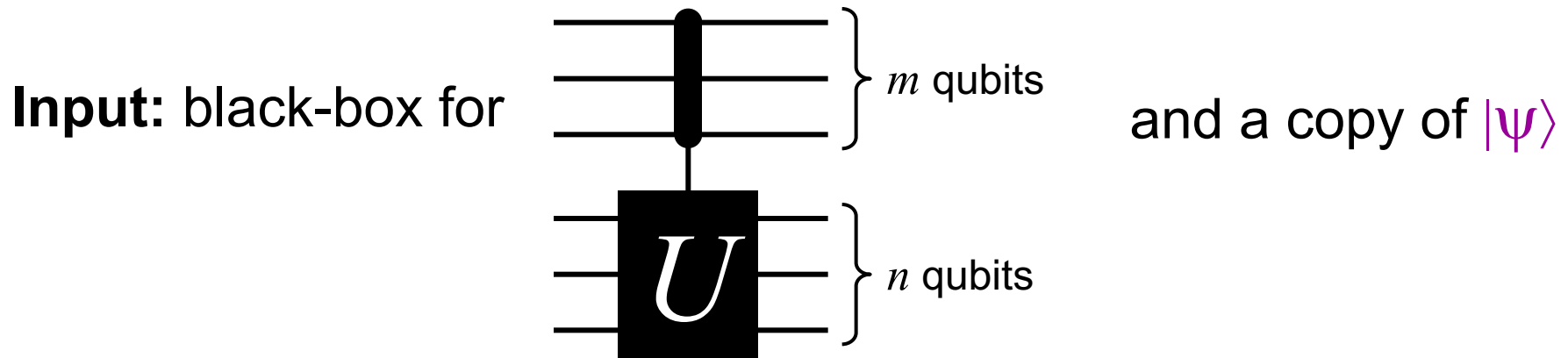
$$\begin{bmatrix} I & 0 & 0 & \dots & 0 \\ 0 & U & 0 & \dots & 0 \\ 0 & 0 & U^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & U^{2^m - 1} \end{bmatrix}$$

Example: $|1101\rangle|0101\rangle \rightarrow |1101\rangle U^{1101}|0101\rangle$

Eigenvalue estimation problem

U is a unitary operation on n qubits

$|\psi\rangle$ is an eigenvector of U , with eigenvalue $e^{2\pi i\phi}$ ($0 \leq \phi < 1$)



Output: ϕ (m -bit approximation)

- Algorithm:**
- one query to generalized controlled- U gate
 - $O(n^2)$ auxiliary gates
 - Success probability $4/\pi^2 \approx 0.4$

Note: with $2m$ -qubit control gate, error probability is exponentially small

Order-finding via eigenvalue estimation

Order-finding problem

Let m be an n -bit integer (**not** necessarily prime)

Def: $\mathbb{Z}_m^* = \{x \in \{1, 2, \dots, m-1\} : \gcd(x, m) = 1\}$ a group (mult.)

Def: $\text{ord}_m(a)$ is the minimum $r > 0$ such that $a^r = 1 \pmod{m}$

Order-finding problem: given m and $a \in \mathbb{Z}_m^*$ find $\text{ord}_m(a)$

Example: $\mathbb{Z}_{21}^* = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$

The powers of 5 are: 1, 5, 4, 20, 16, 17, 1, 5, 4, 20, 16, 17, 1, 5, ...

Therefore, $\text{ord}_{21}(5) = 6$

Note: no *classical* polynomial-time algorithm is known for this problem—it turns out that this is as hard as factoring

Order-finding algorithm (1)

Define: U (an operation on n qubits) as: $U|y\rangle = |ay \bmod m\rangle$

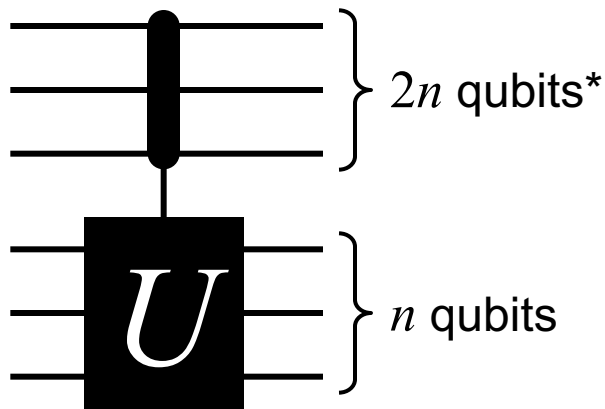
Define: $|\psi_1\rangle = \sum_{j=0}^{r-1} e^{-2\pi i(1/r)j} |a^j \bmod m\rangle$

Then $U|\psi_1\rangle = \sum_{j=0}^{r-1} e^{-2\pi i(1/r)j} |a^{j+1} \bmod m\rangle$
 $= \sum_{j=0}^{r-1} e^{2\pi i(1/r)} e^{-2\pi i(1/r)(j+1)} |a^{j+1} \bmod m\rangle$
 $= e^{2\pi i(1/r)} |\psi_1\rangle$

Therefore $|\psi_1\rangle$ is an eigenvector of U

And knowing the eigenvalue is equivalent to knowing $1/r$,
from which r can be determined

Order-finding algorithm (2)



corresponds to the mapping:

$$|x\rangle|y\rangle \mapsto |x\rangle|a^x y \bmod m\rangle$$

Moreover, this mapping can be implemented with $O(n^2 \log n)$ gates (n multiplications in “repeated squaring” algorithm)

The eigenvalue estimation algorithm yields a $2n$ -bit estimate of $1/r$ (using the above mapping and the state $|\psi_1\rangle$)

From this, a good estimate of r can be calculated by taking the reciprocal, and rounding off to the nearest integer

Exercise: why are $2n$ bits necessary and sufficient for this?

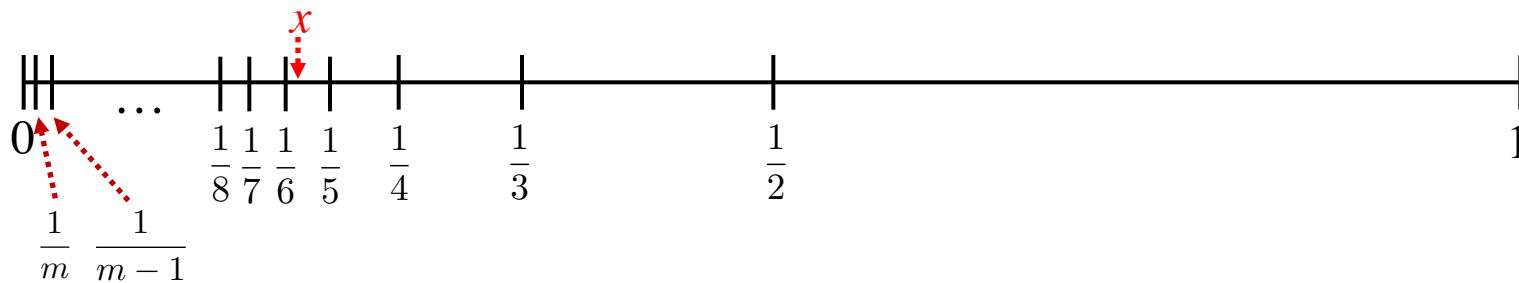
Big problem: how do we construct state $|\psi_1\rangle$ to begin with?

* we're now using m for the modulus and setting the number of control qubits to $2n$

Order-finding algorithm (3)

Solution to exercise: If $r \in \{1, 2, \dots, m\}$, where m is an n -bit integer then a $2n$ -bit approximation of $1/r$ is necessary and sufficient to determine r

The obvious procedure is to check where x lands on the line and round to the nearest $1/r$



The hardest case to distinguish is between $1/m$ vs $1/(m-1)$, where the gap is

$$\frac{1}{m-1} - \frac{1}{m} = \frac{1}{(m-1)m}$$

When $m = 2^n$, this gap is $\frac{1}{(2^n - 1)2^n} \approx \frac{1}{2^{2n}}$

This is the basic idea why $2n$ bits precision is necessary and sufficient

Bypassing the need for $|\psi_1\rangle$ (1)

Note: if we let

$$\begin{aligned} |\psi_1\rangle &= \sum_{j=0}^{r-1} e^{-2\pi i(1/r)j} |a^j \bmod m\rangle \\ |\psi_2\rangle &= \sum_{j=0}^{r-1} e^{-2\pi i(2/r)j} |a^j \bmod m\rangle \\ &\vdots \\ |\psi_k\rangle &= \sum_{j=0}^{r-1} e^{-2\pi i(k/r)j} |a^j \bmod m\rangle \\ &\vdots \\ |\psi_r\rangle &= \sum_{j=0}^{r-1} e^{-2\pi i(r/r)j} |a^j \bmod m\rangle \end{aligned}$$

then **any** one of these could be used in the previous procedure, yielding an estimate of k/r , from which r can be extracted

What if k is chosen randomly and kept secret?

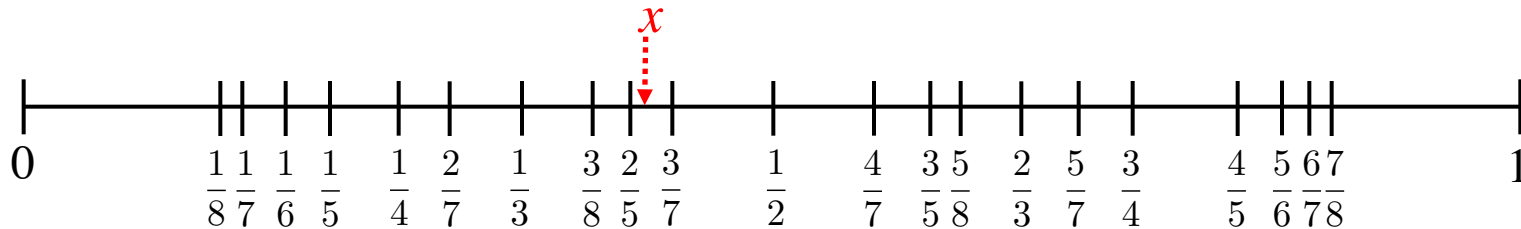
Bypassing the need for $|\psi_1\rangle$ (2)

What if k is chosen randomly and kept secret?

Let $r \in \{1, 2, \dots, m\}$, where m is an n -bit integer and $k \in \{1, 2, \dots, r\}$

Given x , a $2n$ -bit approximation of k/r , can we determine k, r ?

The situation is now more complicated, though in principle we could still imagine checking where x lands on the line and round to the nearest k/r



The hardest case to distinguish is still between $1/m$ vs $1/(m-1)$, where the gap is around $1/m^2$

Of course, we cannot distinguish between these r/k : $1/2 = 2/4 = 3/6 = 4/8$, but at least the procedure makes sense when $\gcd(k, r) = 1$

But: is there an algorithm that finds k and r in time polynomial in n ?

Bypassing the need for $|\psi_1\rangle$ (3)

Let $r \in \{1, 2, \dots, m\}$, where m is an n -bits, $k \in \{1, 2, \dots, r\}$, where $\gcd(k,r)=1$

Question: given x , a $2n$ -bit approximation of k/r , is there an efficient algorithm to determine k, r ? (i.e. by *efficient*, we mean time polynomial in n)

Answer: Yes, the *continued fractions algorithm** does exactly this!

* For a discussion of the *continued fractions algorithm*, please see Appendix A4.4 in [Nielsen & Chuang]

Bypassing the need for $|\psi_1\rangle$ (4)

What is the probability that k and r are relatively prime?

Recall that k is randomly chosen from $\{1, \dots, r\}$

The probability that this occurs is $\phi(r)/r$, where ϕ is **Euler's totient function** (which is defined as the size of \mathbb{Z}_r^*)

It is known that $\phi(r) = \Omega(r/\log\log r)$, which implies that the above probability is at least $\Omega(1/\log\log r) = \Omega(1/\log n)$

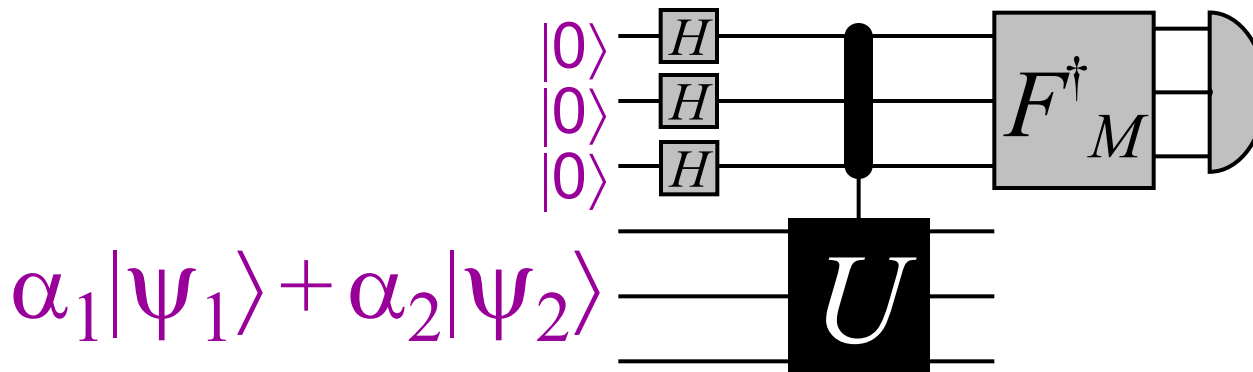
Therefore, the success probability is at least $\Omega(1/\log n)$

Is this good enough? Yes, because it means that the success probability can be amplified to any constant < 1 by repeating $O(\log n)$ times (so still polynomial in n)

But we'd still need to generate a random $|\psi_k\rangle$ here ...

Bypassing the need for $|\psi_1\rangle$ (5)

Returning to the phase estimation problem, suppose that $|\psi_1\rangle$ and $|\psi_2\rangle$ have respective eigenvalues $e^{2\pi i\phi_1}$ and $e^{2\pi i\phi_2}$, and that $\alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle$ is used in place of an eigenvector:



What will the outcome of the measurement be?

It can be shown* that the outcome will be an estimate of

$$\begin{cases} \phi_1 & \text{with probability } |\alpha_1|^2 \\ \phi_2 & \text{with probability } |\alpha_2|^2 \end{cases}$$

* Showing this is straightforward, but not entirely trivial

Bypassing the need for $|\Psi_1\rangle$ (6)

Along these lines, using $\frac{1}{\sqrt{r}} \sum_{k=1}^r |\psi_k\rangle$ yields the same outcome as using a random $|\psi_k\rangle$ (but not being given k), where each $k \in \{1, \dots, r\}$ occurs with probability $1/r$ — this is a case that we've already solved

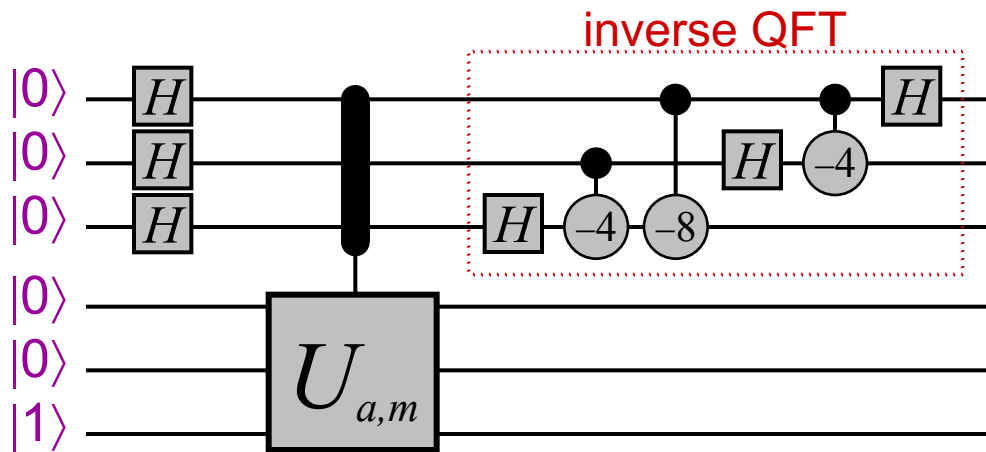
So now all we have to do is construct the state $\frac{1}{\sqrt{r}} \sum_{k=1}^r |\psi_k\rangle$

Since
$$\begin{aligned} \frac{1}{\sqrt{r}} \sum_{k=1}^r |\psi_k\rangle &= \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \left(|1\rangle + \omega^{-1}|a\rangle + \omega^{-2}|a^2\rangle + \dots + \omega^{-(r-1)}|a^{r-1}\rangle \right) \\ &+ \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \left(|1\rangle + \omega^{-2}|a\rangle + \omega^{-4}|a^2\rangle + \dots + \omega^{-2(r-1)}|a^{r-1}\rangle \right) \\ &+ \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \left(|1\rangle + \omega^{-3}|a\rangle + \omega^{-6}|a^2\rangle + \dots + \omega^{-3(r-1)}|a^{r-1}\rangle \right) \\ &+ \vdots \quad \quad \quad \vdots \\ &+ \frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \left(|1\rangle + \omega^{-r}|a\rangle + \omega^{-2r}|a^2\rangle + \dots + \omega^{-r(r-1)}|a^{r-1}\rangle \right) = |1\rangle \end{aligned}$$

its easy!

This is how the previous requirement for $|\Psi_1\rangle$ is bypassed

Quantum algorithm for order-finding



measure these qubits and apply **continued fractions algorithm** to determine (with constant success probability) k' and r' such that $k'/r' = k/r$

$$U_{a,m} |y\rangle = |ay \bmod m\rangle$$

Number of gates for $\Omega(1/\log n)$ success probability is: $O(n^2 \log n)$
(this is the cost of $O(n)$ multiplications)

For any **constant** success probability, repeat $O(\log n)$ times and take the smallest resulting r' that satisfies the equation $a^{r'} = 1 \pmod{m}$

Reduction from factoring to order-finding

The integer factorization problem

Input: m (n -bit integer; we can assume it is composite)

Output: p, q (each greater than 1) such that $pq = m$

Note 1: no efficient (polynomial-time) classical algorithm is known for this problem

Note 2: given any efficient algorithm for the above, we can recursively apply it to fully factor m into primes* efficiently

* A polynomial-time **classical** algorithm for **primality testing** exists

Factoring prime-powers

There is a straightforward *classical* algorithm for factoring numbers of the form $m = p^k$, for some prime p

What is this algorithm?

Therefore, the interesting remaining case is where m has at least two distinct prime factors

Numbers other than prime-powers

Proposed quantum algorithm (repeatedly do):

1. randomly choose $a \in \{2, 3, \dots, m-1\}$
2. compute $g = \gcd(a, m)$
3. **if $g > 1$ then**
 output $g, m/g$
else
 compute $r = \text{ord}_m(a)$ (quantum part)
 if r is even then
 compute $x = a^{r/2} - 1 \pmod m$
 compute $h = \gcd(x, m)$
 if $h > 1$ then output $h, m/h$

Analysis:

we have $m \mid a^r - 1$

so $m \mid (a^{r/2} + 1)(a^{r/2} - 1)$

thus, either $m \mid a^{r/2} + 1$
or $\gcd(a^{r/2} + 1, m)$
is a nontrivial factor of m

It can be shown that at least half of the $a \in \{2, 3, \dots, m-1\}$ have even order and result in $\gcd(a^{r/2} + 1, m)$ being a nontrivial factor of m