Quantum Computation
Shor's algorithm

Luís Soares Barbosa & Renato Neves

Universidade do Minho

HASLab
HIGH-ASSURANCE SOFTWARE LABORATORY

INL
INTERNATIONAL IBERIAN NANOTECHNOLOGY LABORATORY

MSc Physics Engineering

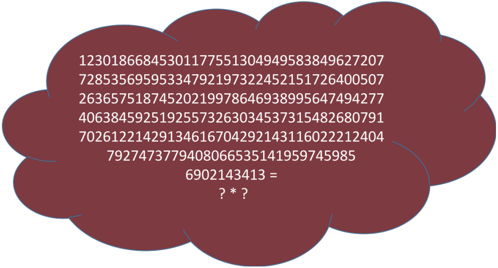
Universidade do Minho, 2024-25

Shor's algorithm

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer

Proc. 35th Annual Symp. on Foundations of Computer Science, IEEE Computer Society Press, pp. 124-134 (1994)

was a turning point in quantum computing for its spectacular decrease of the time complexity of factoring from $\mathcal{O}(e^{\sqrt[3]{n}})$ to $\mathcal{O}(n^3 \log n)$, with potential impact in cryptography.



Factorization

In this famous 1994 paper, Peter Shor proved that it is possible to factor a n-bit number in time that is polynomial to n.

The factorization problem

Given an integer n, find positive integers $p_1, p_2, \cdots, p_m, r_1, r_2, \cdots, r_m$ such that

- Integers $p_1, p_2, \cdots, p_m$ are distinct primes;
- and, $n = p_1^{r_1} \times p_2^{r_2} \times \cdots \times p_m^{r_m}$.

Note that one may assume n to be odd and contain at least two distinct odd prime factors (why?)

Factorization

Since the test for primality can be done classically in polynomial time, the factoring problem can be reduced to $\mathcal{O}(\log n)$ instances of the following problem:

The odd non-prime-power integer splitting problem

Given an odd integer n, with at least two distinct prime factors, compute two integers

$$1 < n_1 < n \quad \text{and} \quad 1 < n_2 < n$$

st $n = n_1 \times n_2$

Factorization

Miller proved in 1975 that this problem reduces probabilistically to another problem whose solution resorts to the eigenvalue estimation problem, already studied.

The order-finding problem

Given two coprime integers a and n, i.e. st $gcd(a, n) = 1$, find the order of a modulo n.

Preliminaries: Modular arithmetic

Consider the group of integers modulo n,

$$\mathbb{Z}_n = (\{0, 1, 2, \cdots, n-1\}, \times_n, 1, \ ^{-1})$$

For two integers x and y we write

$$x \equiv y \pmod n \ \text{ iff } \ \text{rem}\,(x, n) = y$$

or, equivalently, $\text{rem}\,(x-y, n) = 0$, where $\text{rem}\,(a, b)$ is the reminder of the integer division of a by b.

Examples

$5 \equiv 0 \pmod 5$ and $6 \equiv 1 \pmod 5$

Preliminaries: Modular arithmetic

Definition

For co-prime integers $a < n$ the order of $a \pmod{n}$ is the smallest integer $r > 0$ s.t.

$$a^r \equiv 1 \pmod{n}$$

Example

If $n = 5$ the sequence $3^0, 3^1, 3^2, 3^3, 3^4, 3^5, 3^6, \ldots$ leads to the sequence $1, 3, 4, 2, 1, 3, 4, \ldots$. Thus, the

$$\text{order of } 3 \pmod{5} \text{ is } 4$$

Exercise

What is the order of $2 \pmod{11}$?

The problem

The order-finding problem

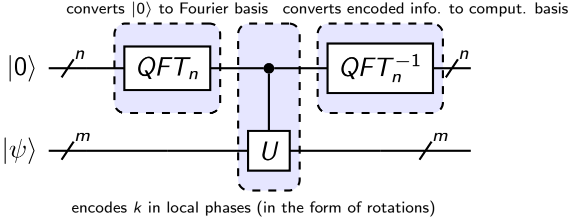
Given two coprime integers a and n, i.e. st $\gcd(a, n) = 1$, find the order of a modulo n, i.e. the smallest positive integer r such that

$$a^r \equiv 1 \pmod n$$

- Classically, this problem can be difficult for large integers.
- In a quantum computer, however, it can be solved efficiently via the quantum eigenvalue estimation algorithm.

Strategy: The eigenvalue approach

Recall the eigenvalue estimation circuit:



Need to choose suitable U and $|\psi\rangle$ to disclose the order

Strategy: The eigenvalue approach

Take co-prime integers $a < n$
Let $m = \lceil \log_2 n \rceil$ and define $U_a : \mathbb{C}^{2^m} \to \mathbb{C}^{2^m}$

$$\begin{aligned} U_a(|q\rangle) &= |\mathrm{rem}\,(qa, n)\rangle \quad \text{for } 0 \leq q < n \\ U_a(|q\rangle) &= |q\rangle \quad \text{for } q \geq n \end{aligned}$$

Exercise

Show U_a is unitary.

Exercise

Show that $U_a\,|\mathrm{rem}\,(a^n, n)\rangle = \left|\mathrm{rem}\,(a^{n+1}, n)\right\rangle$

Next step is to identify suitable eigenvectors.

A first attempt (starting with an axample)

For $n = 5$, sequence

$$3^0, 3^1, 3^2, 3^3, 3^4, 3^5, 3^6, \ldots$$

leads to $\underline{1, 3, 4, 2}, 1, 3, 4, \ldots,$ thus the order r of 3 $\pmod 5$ is 4.

Thus, compute

$$\begin{aligned}
&U_a\Big(\tfrac{1}{\sqrt{r}}(|1\rangle + |3\rangle + |4\rangle + |2\rangle\Big) \\
&= U_a\Big(\tfrac{1}{\sqrt{r}} \textstyle\sum_{i=0}^{r-1} \big|\mathrm{rem}\,(3^i, 5)\big\rangle\Big) \\
&= \tfrac{1}{\sqrt{r}} \textstyle\sum_{i=0}^{r-1} \big|\mathrm{rem}\,(3^{i+1}, 5)\big\rangle \\
&= \tfrac{1}{\sqrt{r}}\Big(|3\rangle + |4\rangle + |2\rangle + |1\rangle\Big) \\
&= \tfrac{1}{\sqrt{r}}\Big(|1\rangle + |3\rangle + |4\rangle + |2\rangle\Big)
\end{aligned}$$

... to conclude that his state is an eigenvector of U_a

A second attempt

The previous example resorts to the equation

$$U_a\left(\frac{1}{\sqrt{r}}\sum_{i=0}^{r-1}\left|\text{rem}\,(a^i,n)\right\rangle\right)=\frac{1}{\sqrt{r}}\sum_{i=0}^{r-1}\left|\text{rem}\,(a^i,n)\right\rangle\right)$$

Unfortunately, the corresponding eigenvalue is 1 ...
... which does not disclose any information about r!

Need to find eigenvectors with more informative eigenvalues.

A second attempt

Since $a^r = 1 \pmod{n}$,

$$U_a^r(|q\rangle) = |\text{rem}(qa^r, n)\rangle = |q\rangle$$

i.e. U_a is the rth-root of the identity operator I, i.e. $(U_a)^r = I$.

It can be shown that the eigenvalues λ of such an operator satisfy

$$\lambda^r = 1$$

i.e. they are rth-roots of 1, which means they take the form

$$e^{i2\pi \frac{k}{r}}$$

for some integer k. In the previous example,

$$1 = e^{i2\pi \frac{0}{r}}$$

A second attempt

Let us consider a different state:

$$|\psi_1\rangle \;=\; \frac{1}{\sqrt{r}} \sum_{i=0}^{r-1} \omega^{-i} \left|\mathrm{rem}\left(a^i, n\right)\right\rangle$$

where $\omega = e^{i2\pi \cdot \frac{1}{r}}$ $\underbrace{\text{(division of the unit circle in } r \text{ slices)}}_{\text{a.k.a. the } r\text{th-roots of unity}}$

$$
\begin{aligned}
& U_a\left(\tfrac{1}{\sqrt{r}} \textstyle\sum_{i=0}^{r-1} \omega^{-i} \left|\mathrm{rem}\left(a^i, n\right)\right\rangle \right) \\
&= \tfrac{1}{\sqrt{r}} \textstyle\sum_{i=0}^{r-1} \omega^{-i} \left|\mathrm{rem}\left(a^{i+1}, n\right)\right\rangle \\
&= \tfrac{1}{\sqrt{r}} \textstyle\sum_{i=0}^{r-1} \omega\omega^{-(i+1)} \left|\mathrm{rem}\left(a^{i+1}, n\right))\right\rangle \\
&= \omega \left(\tfrac{1}{\sqrt{r}} \textstyle\sum_{i=0}^{r-1} \omega^{-(i+1)} \left|\mathrm{rem}\left(a^{i+1}, n\right)\right\rangle \right) \\
&= \omega \left(\tfrac{1}{\sqrt{r}} \textstyle\sum_{i=0}^{r-1} \omega^{-i} \left|\mathrm{rem}\left(a^{i}, n\right)\right\rangle \right)
\end{aligned}
$$

A second attempt

The calculation in the previous slide shows that

$$U_a |\psi_1\rangle = \omega |\psi_1\rangle$$

So if we feed the quantum eigenvalue estimation circuit with U_a and $|\psi_1\rangle$ we obtain an approximation of

$$\frac{1}{r}$$

with a good success probability ($\geq \frac{4}{\pi^2} \approx 0.4$).

Exercise

Formally justify all the steps in that calculation.

Exercise

Would a similar conclusion pop out if our starting state was

$$|\psi_k\rangle \ = \ \frac{1}{\sqrt{r}} \sum_{i=0}^{r-1} \omega^{-ik} \left|\mathrm{rem}\left(a^i, n\right)\right\rangle$$

A third attempt

However ...

How $|\psi_1\rangle$, or, in general, $|\psi_k\rangle$. can be prepared, without knowing r?

Fortunately, it is not necessary!

Instead of preparing an eigenstate corresponding to an eigenvalue $e^{i2\pi\frac{k}{r}}$ for a randomly selected $k \in \{0, 1, \cdots, r-1\}$, it suffices to prepare a uniform superposition of the eigenstates

Then the eigenvalue estimation algorithm will compute a superposition of these eigenstates entangled with estimates of their eigenvalues.

Thus, when a measurement is performed, the result is an estimate of a random eigenvalue.

Question

How to prepare such a superposition without knowing r?

A third attempt

Define

$$|\psi\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle$$

with $|\psi_k\rangle = \frac{1}{\sqrt{r}} \sum_{i=0}^{r-1} \omega^{-ik} \left|\text{rem}\left(a^i, n\right)\right\rangle$.

Exercise

Show that $U_a |\psi_k\rangle = \omega^k |\psi_k\rangle$.

Now observe that

$$\left|\text{rem}\left(a^i, n\right)\right\rangle \; = \; |1\rangle \;\text{ iff }\; \text{rem}\left(i, r\right) = 0$$

Thus, the amplitude of $|1\rangle$ in the above state is the sum over the terms for which $i = 0$

$$\frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-i2\pi \frac{k}{r} 0} \; = \; \frac{1}{r} \sum_{k=0}^{r-1} 1 \; = \; 1$$

A third attempt

Thus, if the amplitude of $|1\rangle$ is 1, the amplitudes of all other basis states are 0, yielding

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle \;=\; |1\rangle$$

Thus, we defined a superposition of eigenvectors that is equal to $|1\rangle$.

Summing up

Thus, the eigenvalue estimation algorithm maps

$$|0\rangle|1\rangle = |0\rangle\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle\right) = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|0\rangle|u_k\rangle \mapsto \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|\tilde{\phi}_k\rangle|u_k\rangle$$

where each $\left|\tilde{\phi}_k\right\rangle$ is the best n-bit approximation of $\frac{k}{r}$ with probability $\geq \frac{4}{\pi^2}$

But how to extract r from $\left|\tilde{\phi}_k\right\rangle$?

To estimate r one resorts another result in number theory ...

Estimating r

Theorem: Let r be a positive integer, and take integers k_1 to k_2 selected independently and uniformly at random from $\{0, 1, \cdots, r-1\}$. Let c_1, c_2, r_1, r_2 be integers st $\gcd(r1, c1) = \gcd(r2, c2) = 1$ and

$$\frac{k_1}{r} = \frac{c_1}{r_1} \quad \text{and} \quad \frac{k_2}{r} = \frac{c_2}{r_2}$$

Then, $r = \text{lcm}(r_1, r_2)$ with probability at least $\frac{6}{\pi^2}$.

Thus

- To obtain $\frac{c_1}{r_1}$ from $\tilde{\phi}_k$, i.e. the nearest fraction approximating $\frac{k}{r}$ up to some precision dependent on the number of qubits used, one resorts to the continued fractions method.
- As a second pair (c_2, r_2) is needed, the whole algorithm is repeated.

Finally... the algorithm

In order to obtain the order r, proceed with the following steps

1. run the quantum eigenvalue estimation followed by the continued fractions algorithm twice to obtain two reduced fractions $\frac{c_1}{r_1}$ and $\frac{c_2}{r_2}$
2. if $gcd(c_1, c_2) \neq 1$ repeat previous step else set r as the least common multiple of r_1 and r_2
3. if $a^r \pmod N \equiv 1$ output r else go back to step 1

In step 2,

- The probability of $gcd(c_1, c_2) = 1$ is $\geq \frac{1}{4}$. Hence whole algorithm has constant probability of success
- computation of gcd and least common multiple has complexity $O(m^2)$. Hence the whole algorithm must be efficient.

Reducing to order-finding

The odd non-prime-power integer splitting problem

Given an odd integer n, with at least two distinct prime factors, compute two integers

$$1 < n_1 < n \quad \text{and} \quad 1 < n_2 < n$$

st $n = n_1 \times n_2$

Miller proved in 1975 that this problem reduces probabilistically to the order-finding problem, all reductions being classical: only the estimation problem is quantum.

Reduction to order-finding

- To split n, choose randomly, with uniform probability, an integer a and compute its order r such that a and n are coprime (test a from $\{2, 3, \cdots, n-2\}$). If they are not coprime, their greatest common divisor is already a non trivial factor of n.
- If r is even (it will be with at least a probability of 0.5), $a^r - 1$ can be factorized as

$$a^r - 1 \; = \; (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1)$$

- As r is the order of a, n divides $a^r - 1$, which means n must share a factor with $(a^{\frac{r}{2}} - 1)$, or $(a^{\frac{r}{2}} + 1)$, or both.

  This factor can be extracted by the Euclides algorithm which efficiently returns $\gcd(a^r - 1, n)$.

Question

But how can be sure such a factor in non trivial?

Reduction to order-finding

- Clearly n does not divide $(a^{\frac{r}{2}} - 1)$.
  Actually, if $\text{rem}\,(a^{\frac{r}{2}} - 1, n) = 0$, $\frac{r}{2}$, rather than r, would be the order of a.
- However, n may divide $(a^{\frac{r}{2}} + 1)$, i.e. $a^{\frac{r}{2}} = 1 \pmod{n}$ and not share any factor with $(a^{\frac{r}{2}} - 1)$.

Thus, the reduction is probabilistic according to the following

Theorem: Let $n = p_1^{r_1} \times p_2^{r_2} \times \cdots \times p_m^{r_m}$ be the prime factorization of an odd number with $m \geq 2$. Then for a random a, chosen uniformely as before, the probability that its order is even and $a^{\frac{r}{2}} \neq -1 \pmod{n}$ is at least $(1 - \frac{1}{2^m}) \geq \frac{9}{16}$.

For number theoretic results see N. Koblitz. *A Course in Number Theory and Cryptography*, Springer, 1994.

Shor's algorithm

1. Choose $1 \leq a \leq n-1$ randomly.
2. If $\gcd(a, n) > 1$, then return $\gcd(a, n)$.
3. If $\gcd(a, n) = 1$, then use the order-finding algorithm to compute r — the order of a wrt n.
4. If r is odd or $a^{\frac{r}{2}} \equiv -1 \pmod{n}$
   then return to 1.
   else return $\gcd(a^{\frac{r}{2}} - 1, n)$ and $\gcd(a^{\frac{r}{2}} + 1, n)$.

Shor's algorithm

Shor's approach to estimate a random integer multiple of $\frac{1}{r}$ in his original paper was different from the one discussed in this lecture, as an application of the eigenvalue estimation algorithm.

Shor's approach (based on period finding)

- Create a state

$$\sum_{x=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x\rangle |\text{rem}\,(a^x, n)\rangle$$

which is shown to be re-written as

$$\sum_{b=0}^{r-1} \left(\frac{1}{\sqrt{2^n}} \sum_{z=0}^{m_b-1} |zr + b\rangle \right) |\text{rem}\,(a^x, n)\rangle$$

where m_b is the largest integer st $(m_b-1)r + b \leq 2^n - 1$.

Shor's algorithm

Shor's approach (based on period finding)

- Measuring the target register yields $\text{rem}\,(a^b, n)$ for b chosen uniformly at random from $\{0, 1, 2, \cdots, r-1\}$, and leaves the control register in

$$\frac{1}{\sqrt{m_b}} \sum_{z=0}^{m_b - 1} |zr + b\rangle$$

- Apply $QFT_{2^n}^{-1}$ to the control register

  Note that, if r, m_b were known (!), applying $QFT_{m_b r}^{-1}$ would lead to

$$\sum_{j=0}^{r-1} e^{-i2\pi \frac{b}{r} j} |m_b j\rangle$$

  i.e. only values x such that $\frac{x}{r m_b} = \frac{j}{r}$ would be measured.
- Measure x and output $\frac{x}{2^n}$.

Shor's algorithm

Note that in both approaches the circuit is the same.
The only difference is the basis in which the state of the system is analysed:

- the eigenvector basis
- the computational basis in Shor's original algorithm.

Shor's original algorithm is based on the period finding algorithm, which is another application of phase estimation
(see [Nielsen & Chuang, 2010] for a complete account)

In all cases, the underlying quantum component is, of course, the *QFT*.

Quantum algorithms

Recall the overall idea:

engineering quantum effects as computational resources

Classes of algorithms

- Algorithms with superpolynomial speed-up, typically based on the quantum Fourier transform, include Shor's algorithm for prime factorization. The level of resources (qubits) required is not yet currently available.
- Algorithms with quadratic speed-up, typically based on amplitude amplification, as in the variants of Grover's algorithm for unstructured search. Easier to implement in current NISQ technology, typical component of other algorithms.
- Quantum simulation

… and we are done!

Where to look further

- Quantum computation is an extremely young and challenging area, looking for young people either with a theoretical or experimental profile.
  Get in touch if you are interested in pursuing studies/research in the area at UMinho, INESC TEC and INL.
- Follow-up courses next semester on
  - Quantum Logic (calculi and logics for quantum programs)
  - Quantum Data Science (algorithms and exciting applications)

Universidade do Minho

HASLab HIGH-ASSURANCE SOFTWARE LABORATORY

INL INTERNATIONAL IBERIAN NANOTECHNOLOGY LABORATORY

Continued Fractions

Method to approximate any real number t with a sequence of rational numbers of the form

$$[a_0, a_1, \cdots, a_p] \text{ defined by } a_0 + \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\cdots + \frac{1}{a_p}}}}$$

computed inductively as follows

$$\begin{aligned} a_0 &= \lfloor t \rfloor & r_0 &= t - a_0 \\ a_j &= \left\lfloor \frac{1}{r_{j-1}} \right\rfloor & r_j &= \frac{1}{r_{j-1}} - \left\lfloor \frac{1}{r_{j-1}} \right\rfloor \end{aligned}$$

The sequence $[a_0, a_1, \cdots, a_p]$ is called the p-convergent of t.
If $r_p = 0$ the continued fraction terminates with a_p and
$t = [a_0, a_1, \cdots, a_p]$,

Continued Fractions

Example: $\frac{47}{13} = [3, 1, 1, 1, 1, 2]$

$$\frac{47}{13} = 3 + \frac{8}{13} = 3 + \frac{1}{\frac{1}{\frac{13}{8}}}$$

$$= 3 + \frac{1}{1 + \frac{5}{8}} = 3 + \frac{1}{1 + \frac{1}{\frac{8}{5}}}$$

$$= 3 + \frac{1}{1 + \frac{1}{1+\frac{3}{5}}} = 3 + \frac{1}{1 + \frac{1}{1+\frac{1}{\frac{5}{3}}}}$$

$$= 3 + \frac{1}{1 + \frac{1}{1+\frac{1}{1+\frac{2}{3}}}} = 3 + \frac{1}{1 + \frac{1}{1+\frac{1}{1+\frac{1}{\frac{3}{2}}}}} = 3 + \frac{1}{1 + \frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}}}$$

Continued Fractions

Theorem: The expansion terminates iff t is a rational number.
[which makes continued fractions the *right*, finite expansion for rational numbers, differently form decimal expansion]

Theorem: $[a_0, a_1, \cdots, a_p] = \frac{p_j}{q_j}$ where

$$\begin{aligned} p_0 &= a_0, \; q_0 = 1 \\ p_1 &= 1 + a_0 a_1 \\ p_j &= a_j p_{j-1} + p_{j-2}, \quad q_j = a_j q_{j-1} + q_{j-2} \end{aligned}$$

Theorem: Let x and $\frac{p}{q}$ be rationals st

$$\left| x - \frac{p}{q} \right| \leq \frac{1}{2q^2}.$$

Then, $\frac{p}{q}$ is a convergent of the continued fraction for x.