Shor's Algorithm for Factorizing Large Integers

G. Eric Moorhouse, UW Math

References

H.-K. Lo, S. Popescu, and T. Spiller, *Introduction to Quantum Computation and Information,* 1998.

C.P. Williams and S.H. Clearwater, *Explorations in Quantum Computing,* 1998.

A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms,* 1974.

P. Shor, 'Quantum computing', proceedings of the International Congress of Mathematicians, 1998.
`http://www.research.att.com/~shor/papers/ICM.pdf`

P. Shor, 'Polynomial-time algorithms for prime factorization and discrete logarithm problems', *SIAM J. Computing* **26** (1997), 1484-1509.
`http://www.research.att.com/~shor/papers/QCjournal.pdf`

The factorization problem

Problem: Given a large integer n (typically several hundred digits long), factorize n as a product of primes.

We will assume (both for simplicity and with a view to RSA cryptanalysis) that $n = pq$ where p and q are large unknown primes. We must determine p and q.

The integers mod n

Let $R = \{0, 1, 2, \ldots, n-1\}$ with addition and multiplication mod n. For $a, b \in R$ we compute

$$a + b \bmod n \quad \text{and} \quad ab \bmod n$$

by first computing the sum or product as an ordinary integer, then taking the remainder upon division by n.

These operations are easily performed in polynomial time in the input size $\ell = \log(n)$ using a classical logical circuit or quantum circuit of size polynomial in ℓ.

For $x \in R$ and $a \geq 0$, the value of

$$x^a \bmod n$$

can also be determined in polynomial time and space.

Example: To compute $x^{183} \bmod n$, first write 183 in binary as 10110111. Then

$$x^{183} = x^{128}x^{32}x^{16}x^{4}x^{2}x^{1}$$

where the powers $x^2, x^4, x^8, \ldots$ are found by successively squaring mod n, then multiplied together (mod n) two at a time only. This way if n has 100 digits, say, then intermediate computations have at most 200 digits.

Reduction of the Factorization Problem

Factorizing n reduces to the following problem:

Given $1 < x < n$, find the *order* of x mod n, i.e. the smallest $r \geq 1$ such that x^r mod n is 1.

Why such an r exists *(almost certainly)*:

The list of powers

$$1, x, x^2, x^3, x^4, x^5, \ldots \pmod{n}$$

must repeat with period $< n$. This period is the order of x mod n since if $x^k = x^j$ then $x^{k-j} = 1$.

Our cancellation of x's above is legitimate assuming x has no factors in common with n. But the probability that x is divisible by p or q is *miniscule.* Moreover in this case p or q is easily found in polynomial time by computing gcd(x, n) using Euclid's Algorithm. In this unlikely event, Shor's algorithm is not necessary.

Problem: Factor the following number.

```
> n:=175179906191667073;
```

$$n := 175179906191667073$$

Solution: First find the order of a randomly chosen x mod n:

```
> x:=372560175302;
```

$$x := 372560175302$$

Our quantum computer gives the order of x mod n as $r = 87589952066302250$:

$$r := 87589952066302250$$

```
> x &^ r mod n;
```

$$1$$

```
> y := x &^ (r/2) mod n;
```

$$y := 67951655829380287$$

The factors of n are:

```
> gcd(y+1,n);
```

$$88917251$$

```
> gcd(y-1,n);
```

$$1970145323$$

This succeeds in factoring n 25% of the time; the remaining 75% of the time we obtain the trivial factors 1 and n.

Discrete Fourier Transform

The *Discrete Fourier Transform* of order q is the unitary matrix

$$U_q = \frac{1}{\sqrt{q}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \zeta & \zeta^2 & \cdots & \zeta^{q-1} \\ 1 & \zeta^2 & \zeta^4 & \cdots & \zeta^{2(q-1)} \\ 1 & \zeta^3 & \zeta^6 & \cdots & \zeta^{3(q-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{q-1} & \zeta^{2(q-1)} & \cdots & \zeta^{(q-1)^2} \end{pmatrix}$$

where $\zeta = e^{2\pi i/q}$.

If q is a product of small prime factors, then U_q can be factored as a product of a small number (polynomial in $\log(q)$) of simpler unitary transformations, each representing the action of a quantum gate acting on only one or two qubits. (E.g. if $q = 2^\ell$ then only $\ell(\ell+1)/2$ such gates are necessary.)

Shor's Algorithm

Given n, find $2n^2 < q < 3n^2$ such that q is a product of small prime factors. We'll suppose $q = 2^\ell$.

Construct a quantum computer with $q^2 = 2^{2\ell}$ qubits (plus additional qubits for 'workspace'). The base states are denoted

$$|a, b\rangle = |a\rangle|b\rangle$$

where a, b are binary vectors (i.e. vectors with entries 0,1) of length ℓ. Equivalently, a and b (called *registers 1 and 2*) are integers $< q$ written in binary.

At any time, the state of the system is given by

$$|\psi\rangle = \sum_{a=0}^{q-1} \sum_{b=0}^{q-1} c_{a,b}|a, b\rangle$$

where

$$c_{a,b} \in \mathbb{C}, \quad \sum_{a,b} |c_{a,b}|^2 = 1$$

and $|c_{a,b}|^2$ is the probability that a measurement of the system will find the state to be $|a, b\rangle$.

Step 1

Prepare the computer in initial state

$$|\psi\rangle = |0, 0\rangle.$$

Then apply the quantum gate

$$R = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

to each of the ℓ qubits in the first register; this leaves the computer in the state

$$|\psi\rangle = \frac{1}{\sqrt{q}}\sum_{a=0}^{q-1} |a\rangle|0\rangle.$$

For example for $q = 2^2$ we have

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & & & & & & & \\ 1 & -1 & & & & & & & \\ & & 1 & 1 & & & & & \\ & & 1 & -1 & & & & & \\ & & & & 1 & 1 & & & \\ & & & & 1 & -1 & & & \\ & & & & & & \ddots & & \\ & & & & & & & 1 & 1 \\ & & & & & & & 1 & -1 \end{bmatrix} \quad \text{(applies } R \text{ to } a_0\text{)}$$

$$\times \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & & 1 & & & & & \\ & 1 & & 1 & & & & \\ 1 & & -1 & & & & & \\ & 1 & & -1 & & & & \\ & & & & \ddots & & & \\ & & & & 1 & & 1 & \\ & & & & & 1 & & 1 \\ & & & & 1 & & -1 & \\ & & & & & 1 & & -1 \end{bmatrix} \text{ (applies } R \text{ to } a_1\text{)}$$

$$\times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{1}{2} (|00,00\rangle + |10,00\rangle + |01,00\rangle + |11,00\rangle)$$

where all vectors have length $q^2 = 16$ and all matrices are 16×16.

Step 2

Fix a randomly chosen x between 1 and n.

Apply the reversible transformation

$$|a, 0\rangle \mapsto |a, x^a \bmod n\rangle$$

to the state of the quantum computer. This transforms the state $|\psi\rangle$ from

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |0\rangle$$

to

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle |x^a \bmod n\rangle.$$

Step 3

Measure the second register only. We observe the second register to be in a base state $|k\rangle$ where k is some power of x mod n (and all powers of x mod n are equally likely to be observed).

This measurement projects the state $|\psi\rangle \in \mathbb{C}^{q^2}$ into the q-dimensional subspace spanned by all base states $|a, k\rangle$ for the fixed k whose value we have observed.

Thus the new state is

$$|\psi\rangle = \frac{1}{\sqrt{M}} \sum_{a \in A} |a, k\rangle$$

where A is the set of all $a < q$ such that x^a mod n is k and $M = |A|$. That is,

$$A = \{a_0,\, a_0{+}r,\, a_0{+}2r,\, \ldots,\, a_0{+}(M{-}1)r\}$$

where $M \approx \frac{q}{r} \gg 1$. Thus

$$|\psi\rangle = \frac{1}{\sqrt{M}} \sum_{d=0}^{M-1} |a_0{+}dr, k\rangle.$$

Step 4

Apply the Discrete Fourier Transform U_q to the first register. This transforms the state from

$$\frac{1}{\sqrt{M}} \sum_{d=0}^{M-1} |a_0 + dr, k\rangle$$

to

$$|\psi\rangle = \frac{1}{\sqrt{qM}} \sum_{c=0}^{q-1} \sum_{d=0}^{M-1} \exp(2\pi i \tfrac{c(a_0+dr)}{q}) |c, k\rangle$$

$$= \sum_{c=0}^{q-1} \frac{e^{2\pi i c a_0 / q}}{\sqrt{qM}} \sum_{d=0}^{M-1} \exp(2\pi i \tfrac{cdr}{q}) |c, k\rangle$$

$$= \sum_{c=0}^{q-1} \frac{e^{2\pi i c a_0 / q}}{\sqrt{qM}} \left(\sum_{d=0}^{M-1} \zeta^d \right) |c, k\rangle$$

where $\zeta = e^{2\pi i c r / q}$.

Step 5

Measure register 1. We observe register 1 to be in state $|c\rangle$ with probability

$$Pr(c) = \frac{1}{qM} \left| \sum_{d=0}^{M-1} \zeta^d \right|^2$$

where $\zeta = e^{2\pi i \frac{cr}{q}}$.

If $\frac{cr}{q}$ is not *very close* to an integer, then powers of ζ very nearly cancel out ('destructive interference') and such states $|c\rangle$ are extremely unlikely to be observed. Note that

$$\sum_{d=0}^{M-1} \zeta^d = \frac{1 - \zeta^M}{1 - \zeta}$$

is small in this case.

But if

$$\frac{cr}{q} \approx d$$

where d is an integer, then $\zeta \approx 1$ and

$$Pr(c) \approx \frac{M}{qM} = \frac{1}{q}$$

is much larger. Thus the observed probability distribution of c is concentrated around values such that

$$\frac{c}{q} \approx \frac{d}{r}$$

where d is an integer.

Step 6

For the observed value of c, we use a classical computer to find fractions d/r very close to c/q, hoping that this will give us the true order r of x mod n.

For this we use the method of continued fractions, computing the convergents d_1/r_1 to c/q for which the denominator $r < n$. Noting that all the fractions

$$\frac{d_1}{r_1}, \frac{2d_1}{2r_1}, \frac{3d_1}{3r_1}, \ldots$$

are close to c/q, it is reasnoable to try small multiples of r_1 as possible values of r. Odlyzko (1996) suggests trying

$$r_1,\ 2r_1,\ 3r_1, \ldots, \lfloor \log(n)^{1+\epsilon} \rfloor r_1$$

as possible values for r, checking whether x^r mod n gives 1 in each case, and repeating the experiment as often as necessary ($O(1)$ times on average, compared with $O(\log\log n)$ trials on average if multiples of r_1 are not considered).

Example

We simulate a quantum computer attempting to factor $n = 55$. This leads to $q = 2^{13} = 8192$. Let's fix $x = 13$. (This happens to have order $r = 20$.)

Step 1: Initial state.

$$|\psi\rangle = \frac{1}{\sqrt{8192}}\Big(|0,0\rangle + |1,0\rangle + |2,0\rangle + \cdots + |8191,0\rangle\Big)$$

Step 2: Apply modular exponentiation.

$$|\psi\rangle = \frac{1}{\sqrt{8192}}\Big(|0,1\rangle + |1,13\rangle + |2,13^2 \bmod 55\rangle + \cdots + |8191,13^{8191} \bmod 55\rangle\Big)$$

$$= \frac{1}{\sqrt{8192}}\Big(|0,1\rangle + |1,13\rangle + |2,4\rangle + \cdots + |8191,2\rangle\Big)$$

Step 3: Observe register 2.

All ten powers of x mod 55 are equally likely to be observed. Suppose we observe 28 as a power of x mod 55.

$$|\psi\rangle = \frac{1}{\sqrt{410}}\Big(|9,28\rangle + |29,28\rangle + |49,28\rangle + \cdots + |8189,28\rangle\Big)$$

Step 4: Discrete Fourier Transform of register 1.

$$|\psi\rangle = \sum_{c=0}^{8191} \frac{e^{2\pi i\cdot 9c/8192}}{\sqrt{3358720}} \left(\sum_{d=0}^{409} \zeta^d\right) |c,28\rangle$$

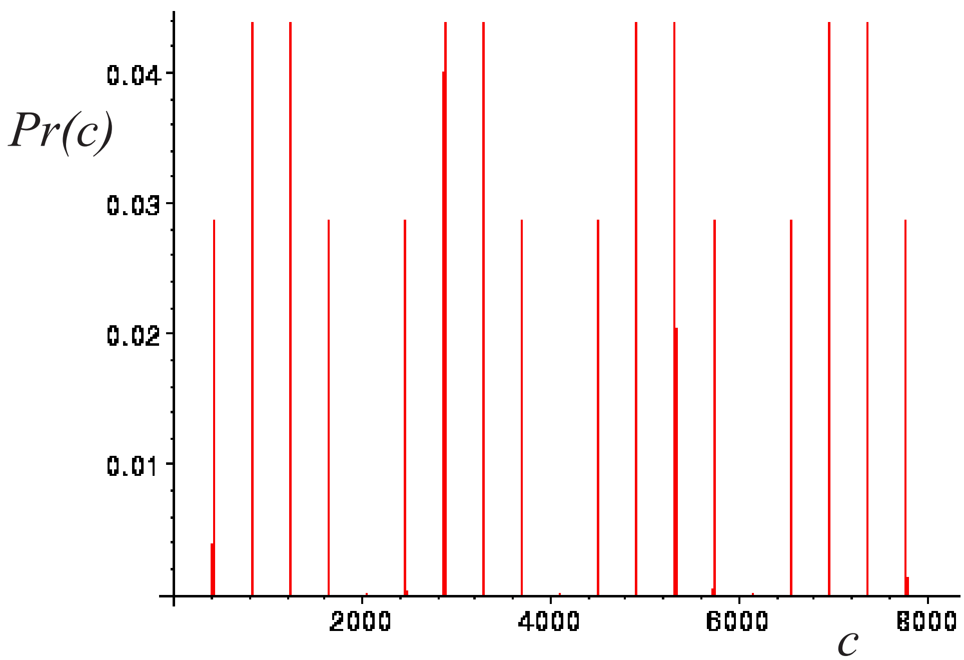
where $\zeta = e^{2\pi i\cdot 20c/8192}$.

Step 5: Measure register 1.

The probability of observing register 1 to be in state $|c\rangle$ is

$$Pr(c) = \frac{1}{3358720} \left| \sum_{d=0}^{409} \zeta^d \right|^2$$

Let's say we observe register 1 to be in state $|4915\rangle$. (This happens with probability 4.4%.)



Step 6: Continued Fraction Convergents

$$\frac{c}{q} = \frac{4915}{8192} = \frac{1}{1 + \frac{1}{1+\frac{1}{2+\frac{1}{1638}}}}$$

Convergents:

$$\frac{1}{1} = 1$$

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

$$\frac{1}{1 + \frac{1}{1+\frac{1}{2}}} = \frac{3}{5}$$

$$\frac{1}{1 + \frac{1}{1+\frac{1}{2+\frac{1}{1638}}}} = \frac{4915}{8192}$$

We stop before the denominator exceeds $n = 55$:

$$r_1 = 5$$

Possible values for r are multiples of $r_1 = 5$:

a	$13^a \bmod 55$
5	43
10	34
15	32
20	1

Evidently $r = 20$. Now

$$y = 13^{10} \bmod 55 \; = \; 34$$

and the factors of $n = 55$ are

$$p = \gcd(y + 1, n) = \gcd(35, 55) = 5;$$
$$q = \gcd(y - 1, n) = \gcd(33, 55) = 11.$$