Shor's Prime Factorization Algorithm

Bay Area Quantum Computing Meetup - 08/17/2017

Harley Patton

Outline

- Why is factorization important?
- Shor's Algorithm
  - Reduction to Order Finding
  - Order Finding Algorithm
- Demo

Why Factorization Matters

- The time it takes for a classical computer to factor some number with *n* digits grows exponentially with *n*, meaning that numbers with many digits take a very long time for a classical computer to factor.
- RSA cryptography and other cryptography algorithms take advantage of this difficulty, and as a result a large amount of information is protected by large semi prime numbers (products of two primes).
- In 1994, mathematician Peter Shor formulated a quantum algorithm to factor an *n*-digit number with a time complexity polynomial in *n*.

Order Finding

- For coprime positive integers *a* and *N*, the order of *a* modulo *N* is defined as the first nonzero *r* such that $a^r = 1$ modulo *N*.
- That's equivalent to finding the period of the following modular exponential function:

$$f(x) = a^x \pmod{N}$$

- This is a problem that can be efficiently solved on a quantum computer.
- Shor realized that if we can find the order *r*, we can use it to quickly factor *N*.

Reducing Factoring to Order Finding

- If integer *a* has order *r* modulo *N*, then:

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

- This can be rewritten as follows:

$$(a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) \equiv 0 \pmod N$$

- Which implies that at least one of the following is a nontrivial factor of *N*:

$$\gcd(a^{\frac{r}{2}} + 1, N)$$

$$\gcd(a^{\frac{r}{2}} - 1, N)$$

- Unless *r* is odd, or the gcd calculation returns *N*.

Example: Factoring 15

- As an example, consider when N = 15 and a = 7. This means we need to find the order of 7 (mod 15), which is the period of the following function:

$$f(x) = 7^x \pmod{15}$$

- Analytically, we can find that the order r is equal to 4:

$$\begin{aligned} f(0) &= 7^0 \pmod{15} = 1 \\ f(1) &= 7^1 \pmod{15} = 7 \\ f(2) &= 7^2 \pmod{15} = 4 \\ f(3) &= 7^3 \pmod{15} = 13 \\ f(4) &= 7^4 \pmod{15} = 1 \end{aligned}$$

- Finally, we can use the order to calculate the factors of 15:

$$\gcd(a^{\frac{r}{2}} - 1, N) \equiv \gcd(7^{\frac{4}{2}} - 1, 15) \equiv 3 \pmod{15}$$
$$\gcd(a^{\frac{r}{2}} + 1, N) \equiv \gcd(7^{\frac{4}{2}} + 1, 15) \equiv 5 \pmod{15}$$

Shor's Algorithm Outline

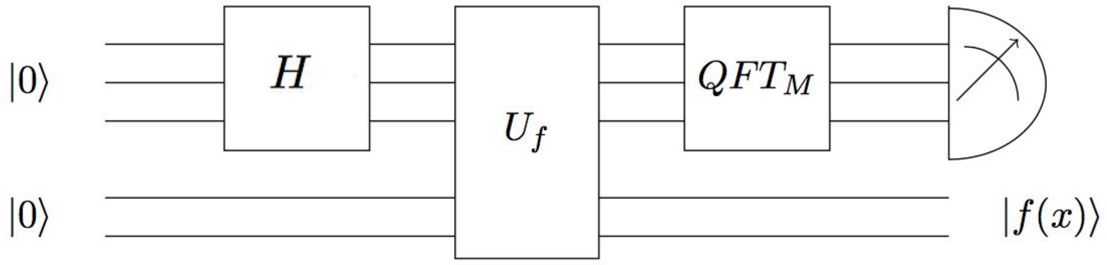
1. Pick a random integer $a < N$
2. If $\gcd(a, N) > 1$, then you have found a nontrivial factor of N.
3. Otherwise, find the order r of a modulo N. (This is the quantum step)
4. If r is odd or a^(r/2) is equivalent to -1 modulo N, go back to step 1.
5. Otherwise, calculate the following values. At least one of them will be a nontrivial factor of N.

$$\gcd(a^{\frac{r}{2}} + 1, N)$$

$$\gcd(a^{\frac{r}{2}} - 1, N)$$

Quantum Circuit for Order Finding

- Quantum circuit for Order Finding acts on two qubit registers, and has three steps:
  a. Use Hadamard transform to put first register into superposition over all bit strings.
  b. Apply unitary function $f(x) = a\hat{}x \pmod{N}$ to second register.
  c. Apply Quantum Fourier Transform to first register and measure.



Step-by-Step Breakdown

- Initially, both registers are in the zero state:

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

- Next, apply Hadamard Transform to first register to create an even superposition over all bit strings *x*:

$$|\psi\rangle = \sum_x |x\rangle\,|0\rangle$$

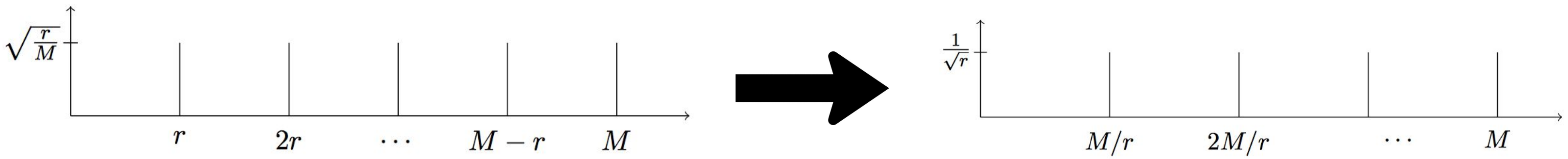
- Then, apply the unitary function f(x) = a^x (mod N) to the second register:

$$|\psi\rangle = \sum_x |x\rangle\,|f(x)\rangle$$

- Finally, apply the QFT and measure the first register. Why?

Fourier Transform Properties

1. **Unitary:** This means the Fourier Transform can be used as a quantum gate.
2. **Period/Wavelength Relationship:** If an *M*-dimensional vector is periodic with period *r*, then its Fourier Transform is only nonzero on multiples of *M*/*r*



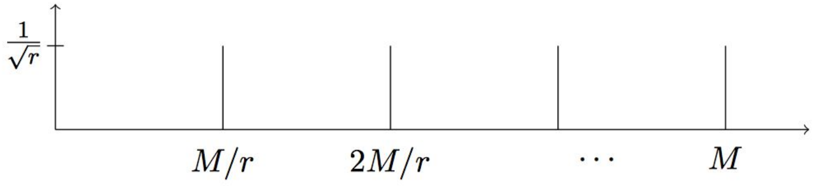
3. **Linear Shift:** Linear shifts of state-vectors cause only phase shifts in their Fourier Transforms.

Explanation of Last Step

- Recall that we had the following wavefunction:

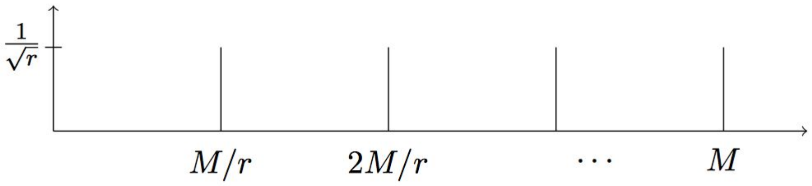
$$|\psi\rangle = \sum_x |x\rangle\, |f(x)\rangle$$

- Suppose that we measured the second register and got a value *y*. The superposition would collapse to only include bitstrings *x* in the first register such that $f(x) = y$
- But this is a periodic function, and so by the **Period/Wavelength Relationship**, its Fourier Transform can only be nonzero on multiples of M/r.
- Because of the **Linear Shift** property, it doesn't even matter what the value *y* obtained was, since after applying the Fourier Transform we end up with the same periodic wavefunction up to a phase:



Extracting the Order

- Recall that we end up with the following wavefunction across the first register:



- Measuring register 1 results in a multiple of M/r. Doing this multiple times and then taking the greatest common denominator of all the results will yield the value M/r with high probability. From there, the order r can be extracted.

iPython Demonstration

- To visualize how this algorithm works, I put together the following demo: http://localhost:8888/notebooks/shor_demo.ipynb