Shor's factoring algorithm: the classical part

Yipeng Huang

Rutgers University

October 6, 2021

Table of contents

Intermediate-term class plan

Where we are headed in first month

1. Fundamentals: superposition / Deutsch-Jozsa
2. Fundamentals: entanglement / Bell inequalities
3. Programming examples in Google Cirq
4. Shor's algorithm (new)
5. A NISQ algorithm: quantum approximate optimization algorithm
6. Programming assignment on QAOA in Cirq

Table of contents

The factoring problem

One way functions for cryptography

1. Multiplying two b-bit numbers: on order of b^2 time.
2. Best known classical algorithm to factor a b-bit number: on order of about $2^{\sqrt[3]{b}}$ time.

- Makes multiplying large primes a candidate one-way function.
- It's an open question of mathematics to prove whether one way functions exist.

Public key cryptography

Numberphile YouTube channel explanation of RSA public key cryptography:
`https://www.youtube.com/watch?v=M7kEpw1tn50`

The factoring problem

One way functions for cryptography

1. Multiplying two b-bit numbers: on order of b^2 time.
2. Best known classical algorithm to factor a b-bit number: on order of about $2^{\sqrt[3]{b}}$ time.

Quantum integer factoring algorithm

- Quantum algorithm to factor a b-bit number: b^3.
- Peter Shor, 1994.
- Important example of quantum algorithm offering exponential speedup.

Table of contents

The classical part: converting factoring to order finding / period finding

General strategy for the classical part

1. Factoring
2. Modular square root
3. Discrete logarithm
4. Order finding
5. Period finding

The fact that a quantum algorithm can support all these primitives leads to additional ways that future quantum computing can be useful / threatening to existing cryptography.

Factoring

$$N = pq$$

$$N = 15 = 3 \times 5$$

Modular square root

Finding the modular square root

$$s^2 \mod N = 1$$

Trivial roots would be $s = \pm 1$.
Are there other roots, and how would it be useful for factoring?

Discrete log

1. Pick a that is relatively prime with N.
2. Efficient to test if relatively prime by finding GCD using Euclid's algorithm.
   For example, a=6 and n=15.

Exercise: list the possible a's for $N = 15$.

Discrete log

1. Pick a that is relatively prime with N.
2. Efficient to test if relatively prime by finding GCD using Euclid's algorithm. For example, a=6 and n=15.

So now our modular square root problem is:

$$a^r \mod N = 1$$

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

In fact, this algorithm for finding discrete log even more directly attacks other crypto primitives such as Diffie-Hellman key exchange.

Order finding

Our discrete log problem is equivalent to order finding.

	$a^1 \mod 15$	$a^2 \mod 15$	$a^3 \mod 15$	$a^4 \mod 15$
a=2	2	4	8	1
a=4	4	1	4	1
a=7	7	4	13	1
a=8	8	4	2	1
a=11	11	1	11	1
a=13	13	4	7	1
a=14	14	1	14	1

Find smallest r such that $a^r \equiv 1 \mod N$

Period finding

In other words, the problem by now can be phrased as finding the period of a function.

$$f(x) = f(x + r)$$

Where

$$f(x) = a^x \mod N$$

Find r.

What to do after quantum algorithm gives you r

- If r is odd or if $a^{\frac{r}{2}} + 1 \equiv 0 \mod N$, abandon.
- There is separate theorem saying no more than a quarter of trials would have to be tossed.

Exercise: try for $a = 14$.

What to do after quantum algorithm gives you r

- If r is odd or if $a^{\frac{r}{2}} + 1 \equiv 0 \mod N$, abandon.
- There is separate theorem saying no more than a quarter of trials would have to be tossed.

Exercise: try for $a = 14$.

Otherwise, factors are GCD($a^{\frac{r}{2}} \pm 1$, N)

a=2	r=4	$2^2 \pm 1 = 4 \pm 1$	
a=4	r=2	$4^1 \pm 1 = 4 \pm 1$	
a=7	r=4	$7^2 \pm 1 = 49 \pm 1$	
a=8	r=4	$8^2 \pm 1 = 64 \pm 1$	Notice this is why we discarded 14.
a=11	r=2	$11^1 \pm 1 = 11 \pm 1$	
a=13	r=4	$13^2 \pm 1 = 169 \pm 1$	
a=14	r=2	$14^2 \pm 1 = 196 \pm 1$	

Proof why this works and why factoring is modular square root

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

So now $a^{\frac{r}{2}}$ is a nontrivial square root of 1 mod N.

$$a^r - 1 \equiv 0 \mod N$$

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

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

is an integer

Proof why this works and why factoring is modular square root

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

is an integer

$\frac{a^{\frac{r}{2}}-1}{N}$ is not an integer

Because that would imply

$$a^{\frac{r}{2}}-1 \equiv 0 \mod N$$

$$a^{\frac{r}{2}} \equiv 1 \mod N$$

but we already defined r is the smallest

$\frac{a^{\frac{r}{2}}+1}{N}$ is not an integer

Because that would imply

$$a^{\frac{r}{2}}+1 \equiv 0 \mod N$$

Which we already eliminated