HRP

Shor's Algorithm ~~and Grover's Algorithm~~

Seungwoo Han

July 15, 2022

Quantum State

Quantum State

Consider a quantum system with n qubits, each of which can have two states—0 and 1. In quantum physics, the state of the quantum system is a point in a 2^n-dimensional vector space.

If we let $|S_i\rangle^a$ to denote basis vectors (or pure states) where i ranges from 0 to $2^n - 1$, a state, or a superposition of the states, is represented by

$$\sum_{i=0}^{2^n-1} a_i |S_i\rangle$$

where $a_i \in \mathbb{C}$ and $\sum_{i=0}^{2^n-1} |a_i|^2 = 1$.

[a] *ket* notation

Probability of Being Observed

Observation

If the machine is observed (w.r.t. the basis), the probability of seeing the state $|S_i\rangle$ is $|a_i|^2$.

We are not going to explore deeply regarding how quantums work in real life, but we shall focus on the mathematical part of quantum system.

Since multiplying by a unit-length complex number does not change the behavior of a state, $2^n - 1$ number of complex numbers are enough to completely describe a state.

Hermitian Conjugate

Hermitian Conjugate

For $A \in \mathrm{Mat}_{m\times n}(\mathbb{C})$, the Hermitian conjugate of A, denoted by $A^*(\in \mathrm{Mat}_{n\times m}(\mathbb{C}))$, is defined by

$$A^* = \overline{A}^{\mathrm{T}} = (\overline{a_{ji}})$$

where a_{ij} denotes the (i, j) entry of A.

Inner Product and Norm

Standard Inner Product

For $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$, which are both column vectors in a n-dimensional complex vector space, the standard inner product is

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^* \mathbf{u}.$$

Note that $\langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{C}$.

Norm

For $\mathbf{v} = (v_1, v_2, \cdots, v_n) \in \mathbb{C}^n$,

$$\|\mathbf{v}\|^2 \triangleq \langle \mathbf{v}, \mathbf{v} \rangle = \sum_{i=1}^{n} |v_i|^2.$$

Unitary Matrix

Unitary Matrix

For $A \in \mathrm{Mat}_{n\times n}(\mathbb{C})$, A is unitary if $A^* = A^{-1}$, i.e.,

$$AA^* = A^*A = I_n,$$

where I_n is the $n \times n$ identity matrix.

For $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$ and a unitary $A \in \mathrm{Mat}_{n\times n}(\mathbb{C})$,

$$\langle A\mathbf{u}, A\mathbf{v}\rangle = \mathbf{v}^*A^*A\mathbf{u} = \mathbf{v}^*\mathbf{u} = \langle \mathbf{u}, \mathbf{v}\rangle.$$

Especially,

$$\|A\mathbf{v}\|^2 = \langle A\mathbf{v}, A\mathbf{v}\rangle = \langle \mathbf{v}, \mathbf{v}\rangle = \|\mathbf{v}\|^2.$$

Changing a State to Another

Changing a State to Another

In order to use a physical system for computation, we must be able to change the state of the system. The laws of quantum mechanics only permit unitary transformations of state vectors.

Physical (or Realistic) Limitations

Quantum circuits (and quantum Turing machines) only allows *local* unitary transformations; that is, unitary transformations on a fixed number of bits. Two-bit transformations can at least in theory be implemented by relatively simple physical systems, while a general n-bit unitary transformation being implemented in real life seems to be in a remote future.
We are not going to investigate the precise definitions of them, but we are going to use polynomially many those two-bit local transformations.

Quantum Gate

Quantum Gate

A quantum gate can be expressed as a (unitary) complex-valued matrix. For instance, a unitary matrix

$$\begin{array}{c|cccc} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ |00\rangle & 1 & 0 & 0 & 0 \\ |01\rangle & 0 & 1 & 0 & 0 \\ |10\rangle & 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ |11\rangle & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{array}$$

corresponds to the transform

$$\begin{aligned} |00\rangle &\longrightarrow |00\rangle \\ |01\rangle &\longrightarrow |01\rangle \\ |10\rangle &\longrightarrow (|10\rangle + |11\rangle)/\sqrt{2} \\ |11\rangle &\longrightarrow (|10\rangle - |11\rangle)/\sqrt{2}. \end{aligned}$$

Quantum Gate Array

Quantum Gate Array

A quantum gate array is a set of quantum gates with logical "wires" connecting their inputs and outputs.

We need to design our circuit *uniformly*

Since there will be a different gate array for each size of input, our circuit design must be *uniform*, i.e.,

- the design of the circuit should be produced by a polynomial-time (classical) computation and
- entries in the unitary matrices describing the gates must be computable numbers.[a]

[a] The first $\log n$ bits of each entry should be classically computable in time polynomial in n.

We Need Reversibility

Why do we need reversibility?

We have to initialize *registers*[a] to prepare the next round of computation. But, manually setting registers to zero is not possible in quantum computers because superpositions of states need to be maintained throughout the whole computation.
Therefore, we need to reverse our computation only with the output, which means we shall make use of completely reversible gate arrays.

[a] where out input-output qubits are stored

Reversible Gate

Reversible Gate

In reversible gates, we can recover the input from the output. Most classical logic gates are irreversible; one cannot recover inputs from an output of the NOT gate.

Toffoli Gate

Toffoli Gate (1980)

Toffoli gate[a] is a reversible gate that is expressed as the following matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Or simply, $(a, b, c) \mapsto (a, b, c \oplus (a \wedge b))$. Note that,

- Toffoli gate is a reversible gate and
- if c is fixed to 1, the third output bit is a `NAND` b.

[a] or sometimes called `CCNOT` (controlled-controlled-not) gate

Classical Circuits Can Be Made Reversible

Toffoli Gate is Universal

Since NAND gate is enough to represent all the other gates[a], i.e., NAND is *universal*, it naturally follows that Toffoli gate is also universal.

[a] NOT: $\neg A = A \texttt{ NAND } A$, OR: $A \vee B = \neg(\neg A \texttt{ NAND } \neg B)$, and so on

Classical Circuits Can Be Made Reversible

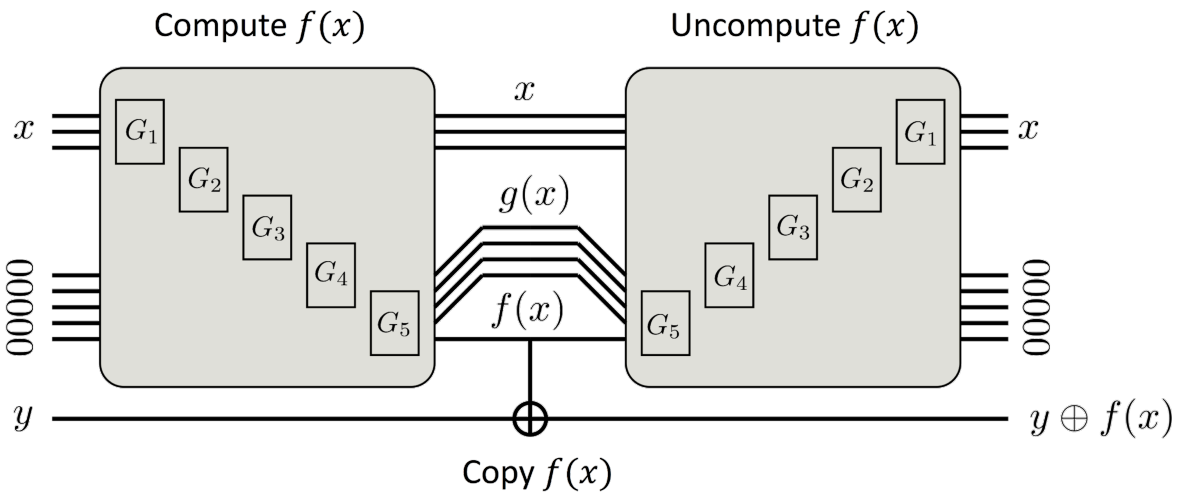
Though it will cost a large amount of space, every classical circuit, including multiplication and modular exponentiation, can be converted into its reversible version (but with extra input and output bits), because Toffoli gate is both reversible and universal.

Bennett's Method

Bennett's Method (1973) ▸ www

We need the input bits to be completely reset; if input bits are not reset after a round of computation, it will affect the next round. This is possible in reversible circuits, and one way is Bennett's method, which is described by the bottom picture, where

- $f(x)$ is the value we want to know, and
- $g(x)$ is the extra output bits used to calculate x from $f(x)$.



Shor's Algorithm

Problem FACTORING:

Given a composite $n \in \mathbb{N}$, find a non-trivial divisor d of n.

Order (Number Theory)

- Given $a, b, n \in \mathbb{Z}$, "$a \equiv_n b$" denotes "$a \equiv b \pmod{n}$", i.e., "$\exists k \in \mathbb{Z},\ a = b + kn$"
- Given $a \in \mathbb{Z}$ and $n \in \mathbb{N}$, "$a \bmod n$" is the unique integer b such that $0 \leq b < n$ and $a \equiv_n b$.

Order

Given $a \in \mathbb{Z}$ and $n \in \mathbb{N}_{>1}$ with $\gcd(a, n) = 1$, the order of a modulo n, denoted by $\mathrm{ord}_n a$, is defined to be

$$\mathrm{ord}_n a \triangleq \min\{i \in \mathbb{N} \mid a^i \equiv_n 1\}$$

The existence is guaranteed by Euler's theorem, $a^{\phi(n)} \equiv_n 1$, where ϕ is Euler's totient function.

Bijection Among $\mathbb{Z}_n^*$s

Let $\mathbb{Z}_n^*$ denote the multiplicative group of integers modulo n. ($|\mathbb{Z}_n^*| = \phi(n)$)

Theorem. (variant of Chinese remainder theorem)

Let $n, n_1, n_2, \cdots, n_k \in \mathbb{N}_{>1}$ are given where $n = \prod_{i=1}^{k} n_i$ and n_i's are pairwise coprime.
Define $f : \mathbb{Z}_n^* \to \mathbb{Z}_{n_1}^* \times \mathbb{Z}_{n_2}^* \times \cdots \times \mathbb{Z}_{n_k}^*$ [a] by

$$f(x) = \left(x \bmod n_1, x \bmod n_2, \cdots, x \bmod n_k\right).$$

Such f is well-defined and is an isomorphism. Moreover,

$$f^{-1}(x_1, x_2, \cdots, x_k) \equiv_n \sum_{i=1}^{k} x_i m_i \left(m_i^{-1} \bmod n_i\right)$$

where $m_i \triangleq N/n_i$.

[a] $\times$ is Cartesian product

Order and Least Common Multiple

Theorem.

Let $n, n_1, n_2, \cdots, n_k \in \mathbb{N}_{>1}$ are given where $n = \prod_{i=1}^{k} n_i$ and n_i's are pairwise coprime. Then,

$$\forall x \in \mathbb{Z}_n^*,\ \operatorname{ord}_n x = \operatorname{lcm}\left(\operatorname{ord}_{n_1} x, \operatorname{ord}_{n_2} x, \cdots, \operatorname{ord}_{n_k} x\right).$$

$$\times_{i=1}^{k} S_i \triangleq S_1 \times S_2 \times \cdots \times S_k$$

Proof.

Let f be the isomorphism in the last page.

$$\begin{aligned}
\operatorname{ord}_n x &= \operatorname{ord}_{\times_{i=1}^{k} \mathbb{Z}_{n_i}^*} f(x) \\
&= \operatorname{ord}_{\times_{i=1}^{k} \mathbb{Z}_{n_i}^*} \left(x \bmod n_1, x \bmod n_2, \cdots, x \bmod n_k\right) \\
&= \operatorname{lcm}\left(\operatorname{ord}_{n_1}(x \bmod n_1), \operatorname{ord}_{n_2}(x \bmod n_2), \cdots, \operatorname{ord}_{n_k}(x \bmod n_k)\right) \\
&= \operatorname{lcm}\left(\operatorname{ord}_{n_1} x, \operatorname{ord}_{n_2} x, \cdots, \operatorname{ord}_{n_k} x\right) \quad \square
\end{aligned}$$

Behavior of $n/\gcd(k,n)$

$$u(p,n) \triangleq \max\left\{k \in \mathbb{Z}_{\geq 0} \;\middle|\; p^k \text{ divides } n\right\} \text{ for each } p \in \mathbb{N}_{>1} \text{ and } n \in \mathbb{N}$$

Theorem.

Let $n \in \mathbb{N}$, $p \in \mathbb{P}$, and $\beta = u(p,n)$. Then,

$$\left|\left\{k \in [n] \;\middle|\; u\big(p, \gcd(k,n)\big) = \alpha\right\}\right| = \begin{cases} \dfrac{p-1}{p^{\alpha+1}} n, & 0 \leq \alpha < \beta \\ \dfrac{1}{p^{\alpha}} n, & \alpha = \beta \end{cases}$$

for each $0 \leq \alpha \leq \beta$, and is 0 for $\alpha > \beta$.

Corollary.

$$\forall \alpha \in \mathbb{Z}_{\geq 0}, \quad \left|\left\{k \in [n] \;\middle|\; u\left(p, \frac{n}{\gcd(k,n)}\right) = \alpha\right\}\right| \leq \frac{p-1}{p} n$$

Order of Powers

Lemma.

Let $n \in \mathbb{N}$ and $a \in \mathbb{Z}$ be integers such that $\gcd(a, n) = 1$ and let $r = \operatorname{ord}_n a$. Then, for any $k \in \mathbb{N}$,

$$\operatorname{ord}_n a^k = \frac{r}{\gcd(k, r)}.$$

Proof.

Let $d \triangleq \gcd(k, r)$. Then, there exist $k_1, r_1 \in \mathbb{N}$ such that $k = k_1 d$ and $r = r_1 d$. Note that $\gcd(k_1, r_1) = 1$. Let $r' = \operatorname{ord}_n a^k$.

From $(a^k)^{r_1} = a^{(k_1 d)(r/d)} = a^{k_1 r} \equiv_n 1$, we get $r' \mid r_1$.

From $a^{kr'} = (a^k)^{r'} \equiv_n 1$, we get $r \mid kr'$, which implies $r_1 \mid k_1 r'$ and thus $r_1 \mid r'$.

Therefore, $r' = r_1 = r/\gcd(k, r)$. □

Probability of Choosing $x \in \mathbb{Z}_{p^\beta}^*$ with $u\left(p, \operatorname{ord}_{p^\beta} x\right) = \alpha$

Theorem.

Let $p \in \mathbb{P} \setminus \{2\}$, $\alpha, \beta \in \mathbb{N}$, and $n = p^\beta$. Then,

$$\Pr_{x \in \mathbb{Z}_n^*}\left[u\left(2, \operatorname{ord}_n x\right) = \alpha\right] \leq \frac{1}{2}.$$

Proof.

Let g be a primitive root[a] of n. Then, for each $x \in \mathbb{Z}_n^*$, there exists $k \in [\phi(n)]$ such that $x \equiv_n g^k$. Therefore, we have

$$\operatorname{ord}_n x = \frac{\operatorname{ord}_n g}{\gcd(k, \operatorname{ord}_n g)} = \frac{\phi(n)}{\gcd(k, \phi(n))}.$$

Since $\phi(n)$ is even, we may conclude that

$$\left|\left\{x \in \mathbb{Z}_n^* \;\middle|\; u\left(2, \operatorname{ord}_n x\right) = \alpha\right\}\right| \leq n/2,$$

and the result follows.

[a] A primitive root is $g \in \mathbb{Z}_n^*$ such that $\operatorname{ord}_n g = \phi(n)$, or equivalently, $\langle g \rangle = \mathbb{Z}_n^*$.

Non-trivial Square Root of 1 Modulo n

Theorem.

If $b \in \mathbb{Z}$ satisfies $b^2 \equiv_n 1$ and $b \not\equiv_n \pm 1$, then

$$1 < \gcd(b-1, n), \gcd(b+1, n) < n.$$

Proof.

Let $d = \gcd(b-1, n)$.

- If $d = 1$, There are $u, v \in \mathbb{Z}$ such that $u(b-1) + vn = 1$. We get $u(b^2-1) + vn(b+1) = b+1$ by multiplying $b+1$ to both sides. As n divides the LHS, it follows that $b \equiv_n -1$.
- Also, d cannot be n because $b - 1 \not\equiv_n 0$.

Similarly, $1 < \gcd(b+1, n) < n$. □

If $r = \mathrm{ord}_n x$ is even and $x^{r/2} \not\equiv_n -1$, then $\gcd(x^{r/2} - 1, n)$ and $\gcd(x^{r/2} + 1, n)$ are non-trivial divisors of n.

Odd Prime Power

Theorem.

If $n = p^k$ where $p \in \mathbb{P} \setminus \{2\}$ and $k \in \mathbb{N}$,

$$m^2 \equiv_n 1 \implies m \equiv_n \pm 1$$

Proof.

From $p^k \mid (m-1)(m+1)$ and $p \in \mathbb{P}$, we get

$$(p^\alpha \mid m-1) \wedge (p^{k-\alpha} \mid m+1)$$

where $0 \leq \alpha \leq k$. It implies $p^{\min(\alpha, k-\alpha)} \mid (m+1)-(m-1) = 2$, or

$$\min(\alpha, k-\alpha) = 0.$$

Therefore, it is either $m \equiv_{p^k} 1$ or $m \equiv_{p^k} -1$. □

A non-trivial divisor of n cannot be found by the method in the last slide if n is an odd prime power.

Asymptotic Growth Rate of Euler's Totient Function

Theorem. (Rosser; 1980) ▸ Paper

For all $n \in \mathbb{N}_{>2}$,

$$\frac{\phi(n)}{n} > \frac{1}{e^{\gamma} \ln\ln n + \frac{3}{\ln\ln n}}$$

where $\gamma = \lim_{m\to\infty} \left(-\ln m + \sum_{k=1}^{m} 1/k\right)$ is Euler's constant.[a]

[a] $\gamma \fallingdotseq 0.577215665$, $e^{\gamma} \fallingdotseq 1.7810724$

Corollary.

$$\frac{\phi(n)}{n} \in \Omega\left(\frac{1}{\log\log n}\right)$$

where Ω is the big-Omega notation.

Continued Fraction

Continued Fraction

For $a_0 \in \mathbb{Z}$, $a_1, \cdots, a_n \in \mathbb{N}$, $a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{\ddots + \cfrac{1}{a_k}}}}$ is called a finite simple continued fraction and is denoted by $[a_0; a_1, a_2, \cdots, a_n]$.

Theorem.

For all $x \in \mathbb{Q} \setminus \mathbb{Z}$, there exist $a_0 \in \mathbb{Z}$ and $a_1, \cdots, a_n \in \mathbb{N}$ such that $x = [a_0; a_1, a_2, \cdots, a_n]$.

Continued Fraction

Convergent

Given a finite simple continued fraction $[a_0; a_1, a_2, \cdots, a_n]$, the k^{th} convergent is

$$C_k = \begin{cases} a_0, & k = 0 \\ [a_0; a_1, \cdots, a_k], & 0 < k \leq n \end{cases}.$$

Also denote $C_k = p_k/q_k$ where $p_k, q_k \in \mathbb{Z}$ such that $\gcd(p_k, q_k) = 1$.

Theorem.

Let $a, p \in \mathbb{Z}$ and $b, q \in \mathbb{N}$ such that $q > b$, $\gcd(a, b) = \gcd(p, q) = 1$.

If $\left|\frac{p}{q} - \frac{a}{b}\right| < \frac{1}{2b^2}$, then $\frac{a}{b}$ is a convergent of $\frac{p}{q}$.

Overview

Given $n \in \mathbb{N}_{>1}$, find a non-trivial divisor d of n if n is composite.

Procedure of Shor's Algorithm

Input: $n \in \mathbb{N}_{>1}$

1. If n is even or is a prime power, then done.
2. Pick a random integer $1 < x < n$ and loop.
   1. If $\gcd(x, n) > 1$, then done.
   2. Calculate $r = \mathrm{ord}_n x$. ▷ where we need quantum computing
   3. If r is even and $x^{r/2} \not\equiv_n -1$, break the loop.
3. $\gcd(x^{r/2} + 1, n)$ and $\gcd(x^{r/2} - 1, n)$ are non-trivial divisors.

Determining Whether $n = p^k$

Check If n Is an Odd Prime Power

If $n = p^k$ where $p \in \mathbb{P} \setminus \{2\}$ and $k \in \mathbb{N}$, then $\sqrt[k]{n}$ must be a prime. Moreover, $\sqrt[k]{n} \geq 3$, which implies $k \leq \log_3 n$. Thus what we have to do is calculating $\sqrt[i]{n}$ and check if $\sqrt[i]{n} \in \mathbb{P}$ for $1 \leq i \leq \lfloor \log_3 n \rfloor$.

Calculating $\sqrt[i]{n}$

$f(x) = x^i - n$ works nicely, with Newton's method, approximating $\sqrt[i]{n}$. In other words, it takes $O(\log n)$ time to get q, the nearest integer of $\sqrt[i]{n}$. We now check if $q \in \mathbb{P}$ and $n = q^i$.

Determining Whether $n = p^k$

PRIMES is in P (Agrawal, Kayal, Saxena; 2002) ▸ Paper

The decision problem that asks if $n \in \mathbb{P}$ for $n \in \mathbb{N}$ (i.e., PRIMES) can be solved in a polynomial time complexity.

"This scheme will thus work
as long as n is odd and not a prime power;
finding factors of prime powers can be done
efficiently with classical methods."

How in 1994?[a]

- Shor

[a] Maybe he meant probabilistic tests such as Miller-Rabin.

Probability of Selecting a "Good" x

$$[n] \triangleq \{1, 2, \cdots, n\} \text{ for each } n \in \mathbb{N}$$

Lemma.

Let $n = \prod_{i=1}^{k} p_i^{a_i}$ where $p_i \in \mathbb{P}$ and $a_i \in \mathbb{N}$. Take any $x \in \mathbb{Z}_n^*$ and let $r \triangleq \operatorname{ord}_n x$ and $r_i \triangleq \operatorname{ord}_{p_i^{a_i}} x$ for each $i \in [n]$. Then, given that r is even, $x^{r/2} \equiv_n -1$ if and only if

$$u(2, r_1) = u(2, r_2) = \cdots = u(2, r_k) > 0.$$

Proof.

Since $r = \operatorname{lcm}(r_1, r_2, \cdots, r_k)$, $\forall i \in [n],\ \exists s_i \in \mathbb{N},\ r = r_i s_i$.

$$\begin{aligned} x^{r/2} \equiv_n -1 &\iff \forall i \in [n],\ x^{r/2} \equiv -1 \pmod{p_i^{a_i}} \\ &\iff \forall i \in [n],\ x^{r_i s_i/2} \equiv -1 \pmod{p_i^{a_i}} \end{aligned}$$

Continued . . .

Probability of Selecting a "Good" x

$$x^{r/2} \equiv_n -1 \iff \forall i \in [n],\ x^{r_i s_i/2} \equiv -1 \pmod{p_i^{a_i}}$$
$$\forall i \in [n],\ \exists s_i \in \mathbb{N},\ r = r_i s_i$$

Since $s_i \equiv_2 0$ implies $x^{r/2} = x^{r_i s_i/2} \equiv 1 \pmod{p_i^{a_i}}$, $\underline{\forall i \in [n],\ s_i \equiv_2 1}$. It implies $\forall i \in [n]\ r_i \equiv_2 0$ since r is even.

Let $r_i = 2^{\alpha_i} t_i$ and $r_j = 2^{\alpha_j} t_j$ for $1 \leq i \neq j \leq n$ where $\alpha_i, \alpha_j \in \mathbb{Z}_{\geq 0}$ and t_i, t_j are odd integers.[a] Assume $\alpha_i < \alpha_j$. Since $2^{\alpha_j} \mid r = r_i s_i = 2^{\alpha_i} t_i s_i$, s_i must be even, which is a contradiction.

Therefore, $u(2, r_i) = \alpha$ for all $i \in [n]$ for some fixed $\alpha \in \mathbb{N}$, proving the "only if" part.

The "if" part is much easier. The fact that $u(2, r) = u(2, r_i)$ for all $i \in [n]$ implies $\forall i \in [n],\ s_i \equiv_2 1$. Therefore, $x^{r_i s_i/2} \equiv (-1)^{s_i} \equiv -1 \pmod{p_i^{a_i}}$ for all $i \in [n]$. □

[a] Note that this is possible only if $k > 1$. It is trivial when $k = 1$.

Probability of Selecting a "Good" x

Theorem.

If n has $k > 1$ prime factors, then the probability of choosing x among $\mathbb{Z}_n^*$ such that $r \equiv_2 0$ and $x^{r/2} \not\equiv_n -1$ is at least $1 - 2^{1-k}$ where $r = \operatorname{ord}_n x$. In other words,

$$\Pr_{x \in \mathbb{Z}_n^*} \left[\left(r \equiv_2 0 \right) \wedge \left(x^{r/2} \not\equiv_n -1 \right) \right]^a \geq 1 - \frac{1}{2^{k-1}}.$$

[a] i.e., the probability of choosing a "good" x

The expected number of trial of choosing x that would lead us to find a non-trivial factor of n is $O(1)$.[a]

[a] Actually, the expected value is less than or equal to 2.

Probability of Selecting a "Good" x

Proof.

Let $n = \prod_{i=1}^{k} p_i^{a_i}$ where $p_i \in \mathbb{P}$ and $a_i \in \mathbb{N}$. Let $f : \mathbb{Z}_n^* \to \times_{i=1}^{k} \mathbb{Z}^*_{p_i^{a^i}}$ be the bijection defined in this page. Therefore, choosing a random $x \in \mathbb{Z}_n^*$ is equivalent to choosing $x_i \in \mathbb{Z}^*_{p_i^{a_i}}$ for each $i \in [n]$ independently.

Note that, by the previous results, the chosen x fails if and only if $u(2, r_i)$s are the same. Therefore, for a choice of $x_1, x_2, \cdots, x_k$ to fail, each x_i should be chosen so that $u(2, r_{i-1}) = u(2, r_i)$, and each probability is at most $1/2$ by the result of this page.

Thus, the probability of choosing a "bad" x is at most $1/2^{k-1}$. □

Modular Exponentiation

Schönhage–Strassen Algorithm (1971) ▸ Wikipedia

Schönhage and Strassen suggested a $\Theta\big(\ell(\log \ell)(\log\log \ell)\big)$ algorithm for multiplying two ℓ-bit integers.

Since the classical square-and-multiply algorithm needs $O(\ell)$ multiplications to calculate $x^a \bmod n$ where x, a, and n are ℓ-bit integers, the total time completely of calculating modular exponentiation is $O\big(\ell^2(\log \ell)(\log\log \ell)\big)$

Modular Exponentiation

A Faster Multiplication Algorithm? ▸ Paper

In March 2019, David Harvey and Joris van der Hoeven announced their discovery of an $O(\ell \log \ell)$ multiplication algorithm. If the algorithm works faultlessly, the time complexity of modular exponentiation, which is the bottleneck of Shor's algorithm, will be reduced to $O(\ell^2 \log \ell)$.

Shor's algorithm

From Wikipedia, the free encyclopedia

Shor's algorithm is a quantum computer algorithm for finding the prime factors of an integer. It was developed in 1994 by the American mathematician Peter Shor.[1]

On a quantum computer, to factor an integer N, Shor's algorithm runs in polynomial time, meaning the time taken is polynomial in $\log N$, the size of the integer given as input.[2] Specifically, it takes quantum gates of orde $O\big((\log N)^2 (\log\log N)(\log\log\log N)\big)$ using fast multiplication,[3] thus demonstrating that the integer factorization problem can be efficiently solved on a quantum computer and is consequently in the complexity

Waiting for this to be updated to $O\big((\log N)^2(\log\log N)\big)$ . . .

Quantum Fourier Transform

Quantum Fourier Transform (QFT)

The quantum Fourier transform is a linear transform that maps a basis vector $|a\rangle$ as

$$|a\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} \omega_N^{ac} |c\rangle$$

where N is the dimension of the vector space and $\omega_N \triangleq e^{2\pi i/N}$.

Note that the quantum Fourier transform is equivalent to multiplying a unitary matrix A_N.

$$\mathrm{QFT}\left(\sum_{a=0}^{N-1} x_a |a\rangle\right) = \frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} x_a \sum_{c=0}^{N-1} \omega_N^{ac} |c\rangle$$

We will now show that QFT can be done with a polynomially many numbers of local quantum gates even if N is exponential.

Constructing a Quantum Gate Array for QFT

Constructing a Quantum Gate Array for QFT

Take $N = 2^\ell$ and let us represent an integer $0 \leq a < N$ in binary as $|a_{\ell-1}a_{\ell-2}\cdots a_0\rangle$. We need only two types of gates: R_j and $S_{j,k}$.

$R_j \in \mathrm{Mat}_{N\times N}(\mathbb{C})$ acts on the j^{th} bit.

$$R_j = \begin{array}{c|cc} & |0\rangle & |1\rangle \\ |0\rangle & 1/\sqrt{2} & 1/\sqrt{2} \\ |1\rangle & 1/\sqrt{2} & -1/\sqrt{2} \end{array}$$

$S_{j,k} \in \mathrm{Mat}_{N\times N}(\mathbb{C})$ acts on the j^{th} and k^{th} bits.

$$S_{j,k} = \begin{array}{c|cccc} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ |00\rangle & 1 & 0 & 0 & 0 \\ |01\rangle & 0 & 1 & 0 & 0 \\ |10\rangle & 0 & 0 & 1 & 0 \\ |11\rangle & 0 & 0 & 0 & e^{i\theta_{k-j}} \end{array}$$

where $\theta_{k-j} \triangleq \pi/2^{k-j}$.

Constructing a Quantum Gate Array for QFT

We apply

$$R_0 S_{0,1} \cdots S_{0,\ell-1} R_1 \cdots R_{\ell-3} S_{\ell-3,\ell-2} S_{\ell-3,\ell-1} R_{\ell-2} S_{\ell-2,\ell-1} R_{\ell-1}$$

to perform a quantum Fourier transform. This will result in

$$|a\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} \omega_N^{ac} |\overline{c}\rangle$$

where $\overline{c}$ is the bit reversal of c.

Later, we may perform a postprocess to recover c from $\overline{c}$.

Only $\frac{\ell(\ell+1)}{2} \in O\big((\log N)^2\big)$ local transformations are used.

Constructing a Quantum Gate Array for QFT

The Matrices Do Perform QFT

We now show $|a\rangle \mapsto N^{-1/2} \sum_{c=0}^{N-1} \omega_N^{ac} |\overline{c}\rangle$ for each basis vector $|a\rangle$.

We shall focus specifically on going from $|a\rangle$ to $|b\rangle$. Let a_j, b_j, and $\overline{b}_j$ denote the j^{th} bit of a, b, and $\overline{b}$, respectively.

We first confirm that the $1/\sqrt{2}$ factors of the ℓ number of R_j matrices multiplies to be $N^{-1/2}$ on the amplitude.

Note that, π is added to the phase if and only if $a_j = b_j = 1$, and θ_{k-j} is added to the phase if and only if $a_j = b_k = 1$.

Therefore, the total phase change is

$$\sum_{0 \le j < \ell} \pi a_j b_j + \sum_{0 \le j < k < \ell} \theta_{k-j} a_j b_k.$$

Constructing a Quantum Gate Array for QFT

$$\sum_{0\le j<\ell} \pi a_j b_j + \sum_{0\le j<k<\ell} \frac{\pi}{2^{k-j}} a_j b_k$$

$$= \sum_{0\le j\le k<\ell} \frac{\pi}{2^{k-j}} a_j b_k$$

$$= \sum_{0\le j\le k<\ell} \frac{\pi}{2^{k-j}} a_j \overline{b}_{\ell-k-1}$$

$$= \sum_{0\le j+k'<\ell} 2\pi \frac{2^j 2^{k'}}{2^\ell} a_j \overline{b}_{k'} \qquad \triangleright \text{ Substitute } k' \to \ell-k-1$$

$$\equiv_{2\pi} \sum_{0\le j,k'<\ell} 2\pi \frac{2^j 2^{k'}}{2^\ell} a_j \overline{b}_{k'} \qquad \triangleright \text{ No effects on adding } 2\pi\text{s}$$

$$= \frac{2\pi}{2^\ell} \sum_{j=0}^{\ell-1} 2^j a_j \sum_{k'=0}^{\ell-1} 2^{k'} \overline{b}_{k'}$$

$$= 2\pi a\overline{b}/N$$

Constructing a Quantum Gate Array for QFT

Therefore, we got

$$|a\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{b=0}^{N-1} e^{2\pi i a\bar{b}/N} |b\rangle = \frac{1}{\sqrt{N}} \sum_{b=0}^{N-1} \omega_N^{a\bar{b}} |b\rangle,$$

and it is immediate that

$$|a\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} \omega_N^{ac} |\bar{c}\rangle$$

since $\bar{\cdot}$ is indeed bijective.

With the postprocess[a], we may get the same result as having

$$|a\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{c=0}^{N-1} \omega_N^{ac} |c\rangle.$$

[a] i.e. reversing the bits after observing

Approximate Quantum Fourier Transform

Approximate QFT (AQFT$_m$) (Coppersmith; 1994) ▸ Paper

If $k-j$ is too large, $\theta_{k-j} = \pi/2^{k-j}$ gets too small and we end up multiplying a very small phase factor; it would be very difficult to do accurately physically.

Parametrized by $m \in [\ell]$, the AQFT$_m$ ignores S_{k-j} if $k-j \geq m$.

Therefore, the total phase change is $\sum_{\ell-m \leq j+k' < \ell} 2\pi \frac{2^j 2^{k'}}{2^\ell} a_j \overline{b}_{k'}$[a].

The difference is $\sum_{0 \leq j+k' < \ell-m} 2\pi \frac{2^j 2^{k'}}{2^\ell} a_j \overline{b}_{k'}$, which is bounded by $2\pi\ell 2^{-m}$.

[a] Follow the procedure in the previous page.

From Initialization to Observation

1. Calculate the Modular Exponentiation

Find $N = 2^\ell$ such that $n^2 \leq N < 2n^2$.

We will construct the quantum gate array for calculating $x^a \bmod n$ that fixes x and n and treats a as the only input. This is possible since we can construct such array in a polynomial time.

Suppose we have the uniformly superpositioned state of two registers[a]

$$N^{-1/2} \sum_{a=0}^{N-1} |a, 0\rangle.$$

Making this is easy, since all we have to do is putting each bit of the first register in the superposition $(|0\rangle + |1\rangle)/\sqrt{2}$.

We now calculate $x^a \bmod n$ via the modular exponentiation algorithm made into a reversible quantum gate array, resulting in

$$N^{-1/2} \sum_{a=0}^{N-1} |a, x^a \bmod n\rangle.$$

[a] storage that contains a (possibly) superpositioned 2^ℓ-bit quantum state.

From Initialization to Observation

2. Apply the Quantum Fourier Transform

$$N^{-1/2} \sum_{a=0}^{N-1} |a, x^a \bmod n\rangle$$

Now we apply the QFT on the first register, mapping $|a\rangle$ to

$$N^{-1/2} \sum_{c=0}^{N-1} \omega_N^{ac} |c\rangle,$$

letting us having the state

$$N^{-1} \sum_{a=0}^{N-1} \sum_{c=0}^{N-1} \omega_N^{ac} |c, x^a \bmod n\rangle,$$

Now, we *observe*.

Probability Analysis

Probability of Observing $|c, x^k \bmod n\rangle$

Let $r = \mathrm{ord}_n x$. We now calculate the probability that our machine ends in a particular state $|c, x^k \bmod n\rangle$ where we may assume $0 \leq k < r$. Summing over all (a, c) to reach the state, we find the probability is

$$\left| N^{-1} \sum_{0 \leq a < N; x^a \equiv_n x^k} \omega_N^{ac} \right|^2.$$

Since $x^a \equiv_n x^k$ if and only if $a \equiv_r k$, the probability is equal to

$$\left| N^{-1} \sum_{b=0}^{m-1} \omega_N^{(k+br)c} \right|^2.$$

where $m - 1 \triangleq \lfloor (N - k - 1)/r \rfloor$. Since ω_N^{kc} is a unit-length constant over the summation, now we have

$$\left| N^{-1} \sum_{b=0}^{m-1} \omega_N^{brc} \right|^2.$$

Probability Analysis

For all $\theta \in \mathbb{R}$, $\left|1-e^{i\theta}\right|^2 = (1-\cos\theta)^2 + \sin^2\theta$
$= 2-2\cos\theta = 4\sin^2(\theta/2)$.

$$\left|N^{-1}\sum_{b=0}^{m-1}\omega_N^{brc}\right|^2 = N^{-2}\left|\frac{1-\omega_N^{mrc}}{1-\omega_N^{rc}}\right|^2$$
$$= N^{-2}\frac{\left|1-\omega_N^{mrc}\right|^2}{\left|1-\omega_N^{rc}\right|^2} = N^{-2}\frac{\sin^2(\pi mrc/N)}{\sin^2(\pi rc/N)}$$

if $\omega_N^{rc} \neq 1$; otherwise it is equal to $N^{-2}m^2$.

Therefore, the probability equals $N^{-2}f(rc/N)$ where $f : \mathbb{R} \to \mathbb{R}$ is defined by $f(x) = \dfrac{\sin^2(\pi mx)}{\sin^2(\pi x)}$ for $x \in \mathbb{R} \setminus \mathbb{Z}$, and $f(x) = m^2$ for $x \in \mathbb{Z}$.

We now analyze $f(x)$ to find the lower bound of the probability.

Probability Analysis

some facts regarding $m = \lfloor (N-k-1)/r \rfloor$

$$m = \left\lfloor \frac{N-k-1}{r} \right\rfloor + 1 > \frac{N-k-1}{r} \geq \frac{N}{r} - 1 \geq \left\lfloor \frac{N}{r} \right\rfloor - 1$$

$$\frac{rm}{2N} = \frac{r}{2N}\left(\left\lfloor \frac{N-k-1}{r} \right\rfloor + 1 \right) \leq \frac{N+r-k-1}{2N} \leq \frac{N+r-1}{2N}$$

$$\frac{rm}{2N} = \frac{r}{2N}\left(\left\lfloor \frac{N-k-1}{r} \right\rfloor + 1 \right) > \frac{N-k-1}{2N} \geq \frac{N-r}{2N}$$

$$\sin^2\left(\frac{\pi r m}{2N}\right) > \sin^2\left(\frac{\pi}{2}\frac{N-r}{N}\right) = \cos^2\left(\frac{\pi r}{2N}\right)$$

$$m = \left\lfloor \frac{N-k-1}{r} \right\rfloor + 1 \leq 2\left\lfloor \frac{N-k-1}{r} \right\rfloor \leq 2\left\lfloor \frac{N}{r} \right\rfloor \leq \frac{2N}{r}$$

Therefore, $r/(2N) \leq 1/m$.

Probability Analysis

Let $\lfloor x \rceil \triangleq \min \operatorname{argmin}_{n\in\mathbb{Z}} |x-n|$.[1]

With some calculation involving derivatives, we get

$$0 < x < 1/m \implies f'(x) < 0.$$

Also because of f being an even function, we conclude that

$$|x| \le \frac{r}{2N} \implies f(x) \ge f\left(\frac{r}{2N}\right) = \frac{\sin^2(\pi mr/2N)}{\sin^2(\pi r/2N)} > \cot^2\left(\frac{\pi r}{2N}\right) > \frac{4N^2}{\pi^2 r^2}.$$

Also, since 1 is a period of f, given that $\left|\frac{rc}{N} - \left\lfloor \frac{rc}{N} \right\rceil\right| \le \frac{r}{2N}$, the probability $N^{-2} f(rc/N)$ is greater than

$$\frac{1}{N^2}\frac{4N^2}{\pi^2 r^2} = \frac{4}{\pi^2 r^2} \in \Omega\left(\frac{1}{r^2}\right).$$

[1] i.e., the closest integer to x; choose the least if x is a half integer.

Probability Analysis

When does $\left|rc/N - \lfloor rc/N \rceil\right| \leq r/(2N)$ hold?

$\left|rc/N - \lfloor rc/N \rceil\right| \leq r/(2N)$ holds if and only if the distance between rc and its cloest multiple of N is not greater than $r/2$, or equivalently,

$$\exists d \in \mathbb{N},\ |rc - dN| \leq r/2.$$

It is also equivalent to

$$\exists d \in \mathbb{N},\ \left|\frac{c}{N} - \frac{d}{r}\right| \leq \frac{1}{2N}.$$

Since $1/(2N) \leq 1/(2n^2) < 1/(2r^2)$, there exists a unique such reduced fraction d/r with $r < n$, and we can obtain it by continued fraction in a polynomial time, by the theorem in this page.

Probability Analysis

How many values of $|c, x^k \bmod n\rangle$ would lead us to find the order?

There are $\phi(r)$ values of d/r—since (d, r)s such that $\gcd(d, r) = 1$ will be found by continued fraction—and each would lead to a unique value of c because

$$\left|\frac{c}{N} - \frac{d}{r}\right| \leq \frac{1}{2N}$$

must hold.

Also, being independent to the value of c, the second register $(x^k \bmod n)$ may have at most r values.

Therefore, we have total $r\phi(r)$ values of $|c, x^k \bmod n\rangle$ that enable us to find r using the method mentioned in the last page.

In other words, there are $r\phi(r)$ values of $|c, x^k \bmod n\rangle$ that satisfies $\left|rc/N - \lfloor rc/N \rceil\right| \leq r/(2N)$.

Probability Analysis

Probability of *Observing* a "Good" $|c, x^k \bmod n\rangle$

There are $r\phi(r)$ values of $|c, x^k \bmod n\rangle$ that satisfies $\left|rc/N - \lfloor rc/N \rceil\right| \leq r/(2N)$, i.e. *good* values.

Moreover, as we previously showed, each of those good values has a probability of $\Omega(1/r^2)$.

So the total probability is

$$r\phi(r) \cdot \Omega\left(\frac{1}{r^2}\right) = \Omega\left(\frac{\phi(r)}{r}\right) = \Omega\left(\frac{1}{\log\log r}\right).$$

Thus, we need $O(\log \ell)$ number of observations to get a good $|c, x^k \bmod n\rangle$.

Optimization Methods

Optimization Methods

Since quantum observations seem to be much more expensive than classical computation methods, we shall minimize the expected number of observations needed to find r.

1. $c \pm 1, c \pm 2, \cdots$ are good candidates if c is observed.
2. The algorithm does not consider if r shares a common factor with d. If they do, the common factor is likely to be small. With considering first $(\log n)^{1+\varepsilon}$ multiples of r' calculated from continued fraction, we may reduce the expected number of observing from $O(\log \ell)$ down to $O(1)$. (Odylzko; 1995)
3. If two candidates of r—r_1 and r_2—are found through a few observations, $\mathrm{lcm}(r_1, r_2)$ is a good candidate. This also reduce the expected number to $O(1)$. (Knill; 1995)[a]

[a] These results are introduced in the paper as a part of "conversation". . .