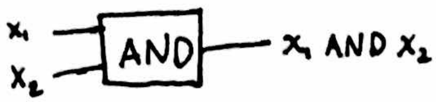
Shor's Algorithm

Quantum Computation Basics (The Quantum Circuit Model)

Classical Circuit:



AND: $00 \mapsto 0$ $\quad 10 \mapsto 0$
$01 \mapsto 0$ $\quad 11 \mapsto 1$

Inputs are either 0 or 1, outputs are 0 or 1. These are called 'bits'.
The 'speed' or 'runtime' is a measure of how many logic gates are used.

Quantum Circuits:

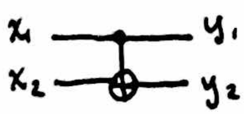
- Instead of bits, use 'qbits': A qbit is $\alpha|0\rangle + \beta|1\rangle \in \mathbb{C}\{|0\rangle, |1\rangle\}$ ← v-space spanned by these two basis vectors.
  Satisfying $|\alpha|^2 + |\beta|^2 = 1$. $\otimes$-product qbits that are adjacent: $|0\rangle \otimes |1\rangle = |01\rangle$.
- All quantum gates are reversible & have the same # of inputs & outputs. They can be thought of as unitary operators on $(\mathbb{C}^2)^{\otimes n}$.
- For us, a quantum computer can perform any unitary operation on one or two qbits. The runtime will measure the # of such small gates.

examples:

Hadamard

$H: |0\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
$|1\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$, extend linearly. eg $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

controlled CNOT



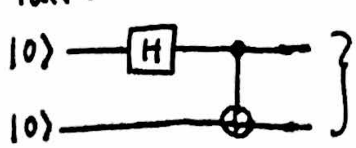
classically,
If $x_1 = 1$, Then $y_2 =$ NOT x_2.
$y_1 = x_1$ always.

quantum,
CNOT: $|00\rangle \mapsto |00\rangle$
$|01\rangle \mapsto |01\rangle$
$|10\rangle \mapsto |11\rangle$
$|11\rangle \mapsto |10\rangle$, extend linearly.

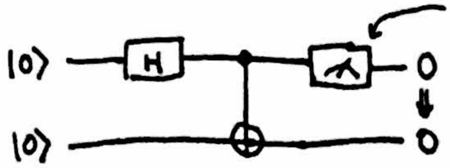
Entanglement & Measurement:

A two-qbit gate can entangle two qbits. Measurement of one qbit determines (or partially collapses the state) the other.

EPR Pair:



output is $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$



measurement of only one qbit determines the other one.

Quantum Fourier Transform

First, let's agree to not care about normalization too much. actually its fine if we do

The state $\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$ is basically the same as $\left(\sum_{x\in\{0,1\}^n} |\alpha_x|^2\right)^{-1}\left(\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle\right)$.

Normally we'd represent a state as above, in the basis $\{|x\rangle : x\in\{0,1\}^n\}$ for $(\mathbb{C}^2)^{\otimes n}$. $(\mathbb{C}^2)^{\otimes n}$ can be thought of as the set of functions from $\{0,1\}^n$ to $\mathbb{C}$. Now think of $\{0,1\}^n$ as $\mathbb{Z}_N$ where $N = 2^n$, by identifying an integer in $\{0,\dots,N-1\}$ with its binary representation.

Another basis for this v-space is the set of characters on $\mathbb{Z}_N$:

(A character is a hom-sm $\mathbb{Z}_N \to \mathbb{C}^\times$. So if χ is one then $\chi(0)=1$, $\chi(k) = \chi(1)^k$, so (since $N=0$) $\chi(1)$ is an N^{th} root of unity.)

$\{\chi_\gamma : \gamma\in\mathbb{Z}_N\}$ where $\chi_\gamma(x) = \omega^{\gamma\cdot x}$, where $\omega = e^{\frac{2\pi i}{N}}$ is a primitive $\sqrt[N]{1}$.

<u>Theorem</u>: $\{\chi_\gamma : \gamma\in\mathbb{Z}_N\}$ is an orthonormal basis for $\mathbb{C}^{\mathbb{Z}_N}$. after normalizing properly

<u>Proof</u>: $\langle \chi_\sigma | \chi_\gamma\rangle = \sum_{x\in\mathbb{Z}_N} \chi_\sigma(x)^* \chi_\gamma(x) = \sum_{x\in\mathbb{Z}_N} \omega^{-\sigma x}\omega^{\gamma x}$

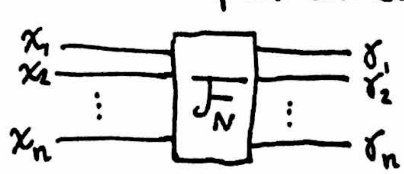
$$= \sum_{x\in\mathbb{Z}_N} \omega^{(\gamma-\sigma)x} = \begin{cases} N & \text{if } \gamma=\sigma \\ 0 & \text{if not}\end{cases}.$$

So we can normalize by $\frac{1}{\sqrt{N}}$ to get a real ONB.

a state $g : \mathbb{Z}_N \to \mathbb{C}$ can be viewed as $\sum_{x\in\mathbb{Z}_N} g(x)|x\rangle$. here we id $|x\rangle$ with 1_x It also has a representation $\sum_{\gamma\in\mathbb{Z}_N} \hat{g}(\gamma)|\chi_\gamma\rangle$ (this defines $\hat{g}$). here we id $|\chi_\gamma\rangle$ with χ_γ

The QFT takes $g \mapsto \hat{g}$. i.e, $\sum_{x\in\mathbb{Z}_N} g(x)|x\rangle = \sum_{\gamma\in\mathbb{Z}_N}\hat{g}(\gamma)|\chi_\gamma\rangle \overset{QFT}{\longmapsto} \sum_{\gamma\in\mathbb{Z}_N}\hat{g}(\gamma)|\gamma\rangle$

This is ~~an ortho~~ a unitary transformation since it takes one ONB to another. The QFT can be implemented with $\binom{n+1}{2} \approx n^2$ simple (1 or 2 qbit) gates. it is denoted like this:



Shor's Algorithm

here $n = \log M \longrightarrow O(n^2)$ or $O(n^6)$ probabilistically / deterministically

Given M, factor it: first, check if it's prime (this can be done quickly)

- find r, a nontrivial square root of 1 mod M (this might not exist if M is a power of an odd prime) (i.e. $r^2 \equiv 1 \bmod M$ but $r \not\equiv \pm 1 \bmod M$).
- Then $(r+1)(r-1) \equiv 0 \bmod M$, but $r+1, r-1 \not\equiv 0 \bmod M$. So both $r+1$ & $r-1$ share a factor with M. Let $c = \gcd(r-1, M)$. (GCD can be computed quickly).
- factor c and $\frac{M}{c}$, return all prime factors. There will be about $\log M$ total recursive calls because M has about $\log M$ prime factors.

How do we find r?

- Pick a random $A \in \mathbb{Z}_M$. compute $\gcd(A, M)$. if it's not 1, then it's a nontrivial factor of M so we've made our algorithm a little faster. if it is 1, then $A \in \mathbb{Z}_M^\times$
- find the order s of $A \in \mathbb{Z}_M^\times$. i.e., $A^s = 1$ but $A^k \neq 1 \ \forall k < s$.
- Suppose we are lucky and s is even. Then $A^{s/2}$ is a square root of 1. Suppose we are even more lucky and $A^{s/2} \neq \pm 1$. Then let $r = A^{s/2}$. It turns out we don't need to try very many times to get this lucky:

<u>Lemma</u>: Suppose M has ≥ 2 distinct odd prime factors. then if we pick $A \in \mathbb{Z}_M^\times$ uniformly at random, $\mathbb{P}(\mathrm{ord}(A)=s \text{ is even } \& \ A^{s/2} \not\equiv 1) \geq \frac{1}{2}$.

So if we try many times and do not find such an A, we can be reasonably sure that M is a power of an odd prime (needless to say M is not even). So we can now binary-search for the k^{th} root of M (which takes $\log M$ time) where $k \in \{1, 2, \ldots, \log M\}$, so in total this will take $(\log M)^2$ time.

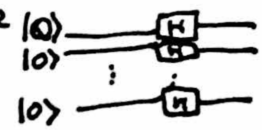
But how do we find $s = \mathrm{ord}(A)$ quickly?

Period-finding algorithm identify $\mathbb{Z}_N = \{0,1\}^n$ where $N = 2^n$.

Problem: Given $f : \mathbb{Z}_N \longrightarrow \{1, \dots, 2^m\}$ = "colors" with the promise that f is periodic — $\exists s \in \mathbb{Z}_N \setminus \{0\}$ s.t. $f(x) = f(x+s)$ $\forall x \in \mathbb{Z}_N$, AND $f(x) \neq f(y)$ whenever $x - y$ is not a multiple of s. Find this s.

Solution Algorithm: Let O_f be an oracle for f : $|x\rangle \otimes |0^m\rangle \longmapsto |x\rangle \otimes |f(x)\rangle$

- prepare the state $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$, which can be done with lots of Hadamard gates:



- attach $|0^m\rangle$ to get $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle \otimes |0^m\rangle$.
- apply the oracle to the state, obtaining $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle \otimes |f(x)\rangle$.
- measure the last m qbits. you'll get a certain color c.
The overall state will collapse to a state $\sum_{x \in \mathbb{Z}_N,\, f(x) = c} |x\rangle \otimes |c\rangle$ normalized.
The normalizing constant is $\sqrt{\frac{s}{N}}$ since there are $\frac{N}{s}$ preimages of c.
We can write this state as $\left(\sum_{x \in \mathbb{Z}_N} f_c(x) |x\rangle \right) \otimes |c\rangle$ where $f_c(x) = \begin{cases} \sqrt{\frac{s}{N}}, & f(x) = c \\ 0, & \text{o.w.} \end{cases}$
- Apply the QFT to the first n qbits, which currently store $f_c \in \mathbb{C}^{\mathbb{Z}_N}$.
Now, let t be the first element of $\mathbb{Z}_N$ for which $f(t) = c$ & $f_c(t) \neq 0$.
Then $f_c(x) = 1_{\{t, t+s, t+2s, \dots\}}(x) \cdot \sqrt{\frac{s}{N}} = \sqrt{\frac{s}{N}} \cdot 1_{\{0, s, 2s, \dots\}}(x-t) = \sqrt{\frac{s}{N}} \sum_{\gamma \text{ a multiple of } \frac{N}{s}} \omega^{\gamma(x-t)} \cdot \frac{1}{s}$
$$= \sum_{\gamma \in \{0, \frac{N}{s}, \frac{2N}{s}, \dots\}} \frac{1}{\sqrt{s}} \omega^{-\gamma t} \chi_\gamma(x) \xrightarrow{\text{QFT}} \sum_{\gamma \in \{0, \frac{N}{s}, \frac{2N}{s}, \dots\}} \frac{\omega^{-\gamma t}}{\sqrt{s}} |\gamma\rangle .$$
Measuring this state yields some $\gamma \in \{0, \frac{N}{s}, \frac{2N}{s}, \dots\}$, and each γ has Probability $\left| \frac{\omega^{-\gamma t}}{\sqrt{s}} \right|^2 = \frac{1}{s}$ of occurring (this is completely ind. of c).
- We can sample from this to obtain a few mulples of $\frac{N}{s}$, then take their GCD. $\gcd(a\frac{N}{s}, b\frac{N}{s}) = \gcd(a,b) \cdot \frac{N}{s}$, so we'll get $\frac{N}{s}$ if a & b are coprime. And the probability of this happening goes to $\frac{6}{\pi^2}$ as N gets large, so for large enough N we don't have to sample too many times. Having $\frac{N}{s}$, divide N by it to get s,

Order-finding Algorithm

Problem: Given m-bit M and $A \in \mathbb{Z}_M^\times$, find $\mathrm{ord}(A)$ in this group. $= s$

Solution Algorithm: let poly(m) be a large polynomial like m^{10}. let $N = 2^{\mathrm{poly}(m)}$.

Define $f : \{0,1,\dots,N-1\} \to \mathbb{Z}_M$ by $f(x) = A^x \bmod M$. $A^0 = A^s = 1$ and all powers in between are distinct, so f is almost s-periodic.

We don't have $s \mid N$, so we modify the period-finding algorithm to fix this.

- Start as before: $\frac{1}{\sqrt{N}} \sum |x\rangle \otimes |0^m\rangle \xrightarrow{O_f} \frac{1}{\sqrt{N}} \sum |x\rangle \otimes |A^x \bmod M\rangle \longmapsto$ measure & collapse.

  We measure a color c, let D be the number of times c occurs, either $\lceil \frac{N}{s} \rceil$ or $\lfloor \frac{N}{s} \rfloor$.

  The collapsed state is $\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |t + s\cdot j\rangle \otimes |c\rangle$ where t is minimal s.t. $A^t \equiv c \bmod M$.

- Apply the QFT. Note that $|x\rangle = 1_x = \sum_{\gamma=0}^{N-1} \chi_\gamma(x)^* \chi_\gamma \xmapsto{QFT} \sum_{\gamma=0}^{N-1} \frac{1}{\sqrt{N}} \omega^{-\gamma\cdot x} |\gamma\rangle$.

  So our collapsed state becomes $\frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} \sum_{\gamma=0}^{N-1} \frac{1}{\sqrt{N}} \omega^{-\gamma t - \gamma\cdot s\cdot j} |\gamma\rangle$, so the probability of measuring a particular γ_0 from this state is $\frac{1}{DN} \left| \sum_{j=0}^{D-1} \omega^{-\gamma_0 t} \omega^{-\gamma_0 s j} \right|^2 = \frac{1}{DN} \left| \sum_{j=0}^{D-1} \omega^{-\gamma_0 s j} \right|^2$.

- we'd like to get a γ such that γs is small mod N. Specifically, if $-\frac{s}{2} \le \gamma s \bmod N \le \frac{s}{2}$ then $|\gamma s - kN| \le \frac{s}{2}$ for some k. equivalently, $\left| \frac{\gamma}{N} - \frac{k}{s} \right| \le \frac{1}{2N}$. So $\frac{\gamma}{N}$ is a good approximation to $\frac{k}{s}$ (better if we take N and poly(m) to be larger).

- Now k was chosen randomly in $\{0,1,\dots,s-1\}$, so with probability at least $\frac{1}{\log s} > \frac{1}{m}$, k and s are coprime. So by computing $\frac{k}{s}$ (using euclid's algorithm on γ and N and stopping when the remainder is small, not 0, i.e. expanding $\frac{\gamma}{N}$ into a continued fraction & stopping early) we can find s.
  (to be sure we have the right s, find two k & k' that give s & are coprime).

- the probability of finding such a γ is positive, at least $\frac{1}{16}$.

  Intuitively, if γs is small then all of $\omega^{-\gamma s j}$ will be close to 1, so they will add positively & not cancel each other out too much.