Lecture 11

Oct 31, 2024

Today: Shor's algorithm & Order factoring

First some notation / discrete math review.

Factoring: given a composite integer $N \in \mathbb{N}$, output integers $1 < a, b < N$ s.t. $ab = N$.

Notation: $a \mid N$ means "a divides N".

$\gcd(a,b) = \max R$ s.t. $R \mid a$ and $R \mid b$.

If $\gcd(a,b) = 1$ then a and b are "coprime".

$\mathbb{Z}_N := $ group over $\{0, \dots, N-1\}$ with idenity 0 and action = addition.

(also denote $\mathbb{Z}/N\mathbb{Z}$).

$\mathbb{Z}_N^\times$ = group over $\{x \in \{1, \dots, N-1\} \mid \gcd(x, N) = 1\}$ with identity 1 and action = multiplication.

Euclid's algorithm for computing $\gcd(a,b)$ with $a \geq b$:

Solve $a = qb + r$ for max q value.

If $r = 0$, $\gcd(a,b) = b$.

Else, $\gcd(a,b) = \gcd(b,r)$.

Runtime: $O(\log a)$ as recursion decreases exponentially fast.

Order finding problem: Input (x, N) s.t. $x \in \mathbb{Z}_N^\times$.

Find minimum r s.t. $x^r \equiv 1 \mod N$. $r = \mathrm{ord}(x)$. Def.

(equiv. find minimum r s.t. x^{r-1} is an inverse of x within $\mathbb{Z}_N^\times$.)

Thm We can reduce the problem of factoring to the order finding problem.

Algorithm for factoring (N):

(Assume N is odd).

1. Sample $1 < x < N$ uniformly at random.
2. Calculate $\gcd(x, N)$. If $\gcd(x, N) \neq 1$, factor found.
3. Use order founding to calculate r s.t.

   $$x^r \equiv 1 \mod N.$$

4. If r is even, compute $\gcd(x^{r/2} - 1, N)$ and $\gcd(x^{r/2} + 1, N)$. If either $\neq 1$, factor found.

5. Repeat until factor is found.

Easy to see that an output is always a factor of N.
Only remains to show that the algorithm only requires a few repetitions till it finds a factor.

In class, we will show today 10 repetitions are needed for 99% confidence if $N = p \cdot q$ for primes p, q.

Lemma Given y s.t. $y^2 \equiv 1 \bmod N$ but $y \not\equiv 1 \bmod N$ and $y \not\equiv -1 \bmod N$, then we can efficiently compute a non-trivial factor of N.

Pf. $y^2 - 1 \equiv 0 \bmod N$. Then,

$(y-1)(y+1) \equiv 0 \bmod N$. Since $y \not\equiv 1$ or $\not\equiv -1 \bmod N$, we know $1 < y < N-1$ and therefore

$\gcd(y-1, N) \neq 1$ or $\gcd(y+1, N) \neq 1$.

Either way, one of the gcd is a divisor of N. Compute both using Euclid's algorithm.

Corollary If we apply it to $y = x^{r/2}$ when $r = \text{ord}(x)$ is even, and $x^{r/2} \not\equiv 1$ or $\not\equiv -1$, this matches step 4.

Next, show that steps 1, 2, 3 sample an x s.t. $r = \text{ord}(x)$ is even and $x^{r/2} \not\equiv 1$ or $\not\equiv -1$ with probability

Fact For prime p, $\mathbb{Z}_p^\times = \{1, 2, \dots, p-1\}$ is a cyclic group.

$\exists\, g \in \mathbb{Z}_p^\times$ s.t. $\{g, g^2, \dots, g^{p-1} = 1\} = \mathbb{Z}_p^\times$. (generator)

Ex. $p = 7$,

5 is a generator: $5,\ 5^2 = 4,\ 5^3 = 6,\ 5^4 = 2,\ 5^5 = 3,\ 5^6 = 1$.

2 isn't a generator: $2,\ 2^2 = 4,\ 2^3 = 1$.

Lemma For odd prime p, choose x uniformly at random from $\mathbb{Z}_p^\times$.

Then, $\mathrm{ord}(x)$ is even with prob. $\geq \frac{1}{2}$.

Pf. Since x is uniformly chosen, $x = g^k$ for random k.

$\frac{1}{2}$ prob, k is odd. Then, $1 \equiv x^{\mathrm{ord}(x)} \equiv g^{k\,\mathrm{ord}(x)}$.

But, $g^{p-1} \equiv 1$. Then either $k\,\mathrm{ord}(x) \mid p-1$ or $p-1 \mid k\,\mathrm{ord}(x)$.

Since k is odd, $p-1$ is even $\Rightarrow$ $\mathrm{ord}(x)$ is even. $\square$

Chinese Remainder Thm (simplified)

If n_1, n_2 are coprime with $N = n_1 n_2$ and

$$\begin{aligned} x &\equiv a_1 \bmod n_1 \\ x &\equiv a_2 \bmod n_2 \end{aligned}$$

has a unique solution $\in \{0, \dots, N-1\}$.

Pf. Suppose x and x' are both solutions. Then $x - x'$ is a multiple of n_1 and $x - x'$ is a multiple of n_2. Since n_1, n_2 coprime $\Rightarrow$ $x - x'$ is a multiple of N.

So, $x = x'$ within $\{0, \dots, N-1\}$. $\square$

In other words, $(a_1, a_2) \mapsto x$ is an injective map.

This is a map $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \to \mathbb{Z}_N$. Since both sets are equal sized, this is a bijection.

Fact $x = a_2 m_1 n_1 + a_1 m_2 n_2$ where m_1, m_2 are integers s.t. $m_1 n_1 + m_2 n_2 = 1$.

Pf.
$$\begin{aligned} x &\equiv a_1 m_2 n_2 \bmod n_1 \\ &\equiv a_1 (1 - m_1 n_1) \bmod n_1 \\ &\equiv a_1 \bmod n_1 \end{aligned}$$

Similarly, $x \equiv a_2 \bmod n_2$.

$(1,1) \mapsto 1$ $\qquad$ $(-1,-1) \mapsto -1$

So $(1,-1)$ and $(-1,1)$ map to values not equal to 1 or -1 by injectivity.

<u>Lemma</u> Let $N = pq$, product of two odd prime numbers. Sample $1 < x < N-1$ uniformly. If $\gcd(x, N) = 1$, Then with prob. $\geq 3/8$, $r = \text{ord}(x)$ is even and $x^{r/2} \not\equiv 1$ or $\not\equiv -1 \mod N$.

<u>Pf.</u> By chinese remainder theorem, sampling $x \in \mathbb{Z}_N$ is equivalent to sampling $(a_1, a_2) \in \mathbb{Z}_p \times \mathbb{Z}_q$. If $a_1 = 0$ or $a_2 = 0$, then x is a multiple of either p or q so $\gcd(x, N) \neq 1$. So, $a_1 \neq 0$ and $a_2 \neq 0$.

Let $r_1 = \text{ord}(a_1)$ within $\mathbb{Z}_p^{\times}$ i.e. min r_1 s.t. $a_1^{r_1} \equiv 1 \mod p$.

$r_2 = \text{ord}(a_2)$ within $\mathbb{Z}_p^{\times}$ i.e. min r_2 s.t. $a_2^{r_2} \equiv 1 \mod q$.

Note, $r_1 | r$ and $r_2 | r$, by def of minimality.

Prev lemma proves r_1 and r_2 are (independently) even with prob. $\geq 1/2$, so r is even with prob. $3/4$. If r is even, then

since $x^r \equiv 1 \mod p$ then $x^{r/2} \equiv 1 \mod p$ or

$x^{r/2} \equiv -1 \mod p$.

Similarly, $x^{r/2} \equiv 1 \mod q$ or $x^{r/2} \equiv -1 \mod q$.

By the previous case work, with half probability

$x^{r/2} \not\equiv 1$ or $\not\equiv -1$.

Overall prob of lemma statement $\geq \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8}$ $\square$

Therefore, factoring reduces to solving order finding.

How to solve order finding:

Input is (x, N) for $x \in \mathbb{Z}_N^\times$.

The goal will be to solve order finding using the same techniques as Simon's or AHSP.

The plus side will be that while both of these were defined w.r.t. an oracle, the oracle gates here can be efficiently implemented.

For $\ell \in \mathbb{Z}^+$, let $f(\ell) = x^\ell \bmod N$ which can be efficiently computed in $\log \ell$ repeated squarings of x classically.

ℓ :	0	1	2	3	...	$r-1$	r	$r+1$	$r+2$	...
$f(\ell)$:	1	x	x^2	x^3		x^{r-1}	1	x	x^2	...

observed periodicity

We want to extract this period — similar to a hidden shift.

Trouble is how do we setup the hidden shift problem?

We could pick a large Q and compute f on $\mathbb{Z}_Q$.

Consider for a moment if we knew Q was a multiple of r.

Then $f(\ell) = f(\ell')$ iff $\ell' - \ell$ is a multiple of r.

So, we can solve r using AHSP algorithm.

Issues:

(1) No easy way of computing Q which was a multiple of r.

(2) We only know an efficient circuit for QFT over $Q = 2^k$.

Resolution: Approximate hidden shift is good enough.

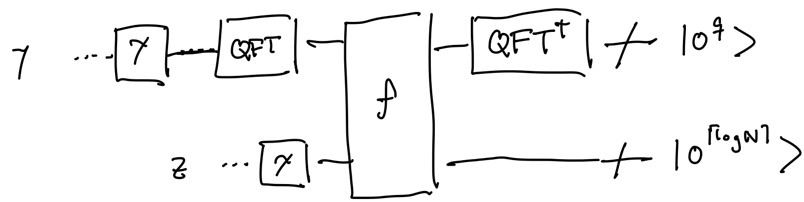
Pick Q as first power of 2 greater than N^2. $(Q = 2^q)$

Now map $\ell \mapsto f(\ell)$ for $\ell \in \mathbb{Z}_Q$ doesn't line-up perfectly but is "mostly correct".

Algorithm:

(1) Generate $\frac{1}{\sqrt{Q}} \sum_{\ell=0}^{Q} |\ell\rangle |f(\ell)\rangle$ and measure second register for output $\in \mathbb{Z}_N$.

② Apply $QFT(\mathbb{Z}_Q)$ and measure to get y.



What values of y are we likely to measure?

Let $\ell, \ell+r, \ell+2r, \ldots, \ell+(J-1)r$ be the sols. to $f^{-1}(z)$.
where $J = \lfloor Q/r \rfloor$.

After collapse to z and QFT_1 state is

$$\frac{1}{\sqrt{J}} \sum_{j=0}^{J-1} QFT\, |\ell + rj\rangle .$$

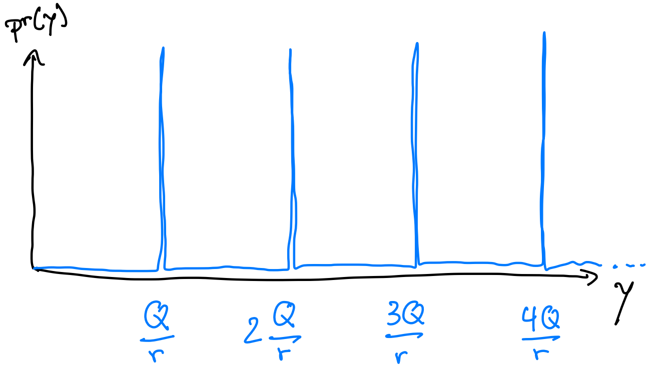
$$= \sum_{y \in \mathbb{Z}_Q} \frac{1}{\sqrt{QJ}} \sum_{j=0}^{J-1} \omega^{y(\ell+rj)} |y\rangle \qquad \text{where } \omega = e^{2\pi i/Q}.$$

$$= \sum_{y \in \mathbb{Z}_Q} \underbrace{\frac{1}{\sqrt{QJ}} \omega^{y\ell} \sum_{j=0}^{J-1} \omega^{yrj}}_{\text{amplitude on } y} |y\rangle$$

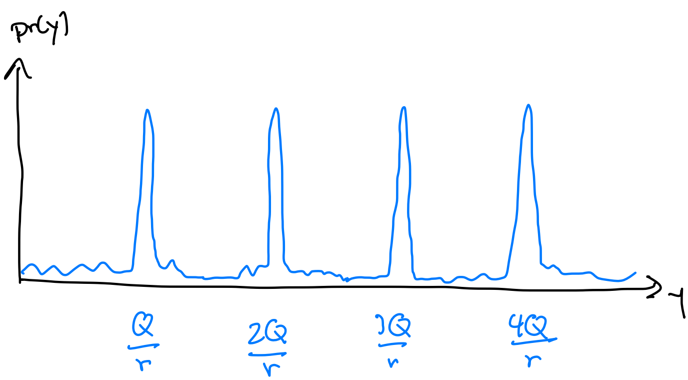
Recall, when $r \mid Q$, only $y = k \cdot \frac{Q}{r}$ are measured

and we calculated GCD of samples y_i to learn Q/r.

Cartoon:



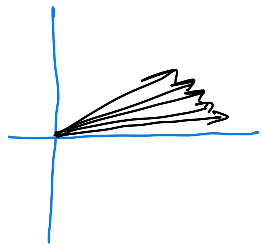
But instead, when $r \nmid Q$, the picture looks more like,



Still peaks at $\frac{Q}{r}$ but is noisy.

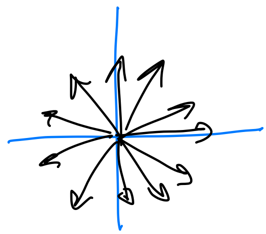
This is b.c. algorithm needs to output an integer but because $r \nmid Q$, constructive and destructive interferences will only mostly align

When $yr \approx 0 \bmod Q$ amplitude sum looks like



constructive interference

When yr far from $0 \bmod Q$ amplitude sum looks like



destructive interference

This time, we will show two ideas:

① With probability $\Omega(1)$, we measure y s.t.

$$-\frac{r}{2} \le yr \bmod Q \le \frac{r}{2}. \quad \text{(constructive interference)}$$

② given such a sample y, we can calculate r, with high prob.

Let's see ② first as it will give us more intuition.

It follows that $\exists\, k \in \mathbb{N}$ s.t.

$$\Rightarrow \quad |yr - kQ| \le \frac{r}{2}.$$

$$\Rightarrow \quad \left|\frac{y}{Q} - \frac{k}{r}\right| \le \frac{1}{2Q}.$$

We know y and we know Q. Reduce fractions to generate

$\frac{y}{Q} = \frac{k'}{r'}$ (both known).

Let's assume $\frac{k'}{r'} \neq \frac{k}{r}$ and show this yields a contradiction.

$$\begin{aligned}
\frac{1}{2Q} &\ge \left|\frac{y}{Q} - \frac{k}{r}\right| \\
&= \left|\frac{k'}{r'} - \frac{k}{r}\right| \\
&= \left|\frac{k'r - r'k}{rr'}\right| \\
&\ge \frac{1}{rr'} \quad \text{(since not equal)} \\
&\ge \frac{1}{N^2}.
\end{aligned}$$

$\Rightarrow \quad N^2 \ge 2Q$ a contradiction since we chose $Q \ge N^2$.

$$\Rightarrow \quad \frac{k'}{r'} = \frac{k}{r}.$$

Let us guess $r = r'$ as the order of x. It is easy to check if $x^{r} \equiv 1 \mod N$.

This guess will be correct if k and r are co-prime.

Note that k is roughly uniformly random $\in \{0, \ldots, r-1\}$, so the probability k and r are coprime is at least

$$\frac{1}{\log k} \geq \frac{1}{\log r} \geq \frac{1}{\log N}.$$

So, with $\geq 1/\log N$ prob., our guess is good.

Only remains to show with prob. $\geq \frac{1}{32}$,

$$-\frac{r}{2} \leq yr \mod Q \leq \frac{r}{2}. \qquad (*)$$

With this, we conclude, we generate $\mathrm{ord}(x)$ with probability, $\Omega(1/\log N)$.

Pf of $(*)$:

Recall amplitude on $|y\rangle$ is $\frac{1}{\sqrt{QJ}} \omega^{y\ell} \sum_{j=0}^{J-1} \omega^{yrj}$

with $J = \lfloor \frac{Q}{r} \rfloor$

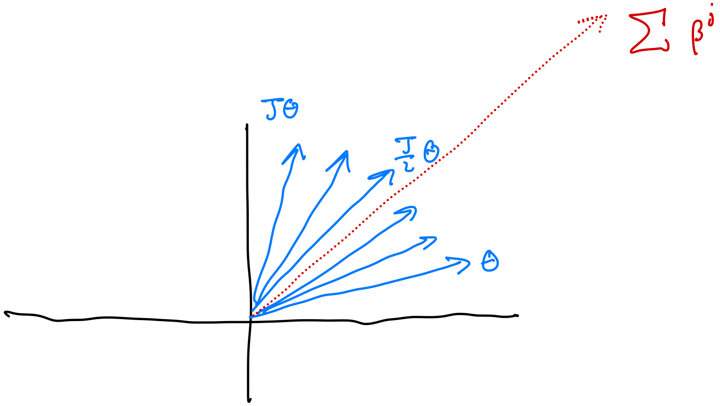
Focus on the $\sum_{j=0}^{J-1} \omega^{yrj}$ term as this is where the constructive/destructive interference occurs. Let $\beta = \omega^{yr} = e^{\left(2\pi i \cdot \frac{yr}{Q}\right)}$

If $\frac{-r}{2} \leq yr \bmod Q \leq \frac{r}{2}$, then $\beta = e^{i\theta}$ for θ an angle s.t. $|\theta| \leq \frac{2\pi r}{2Q} = \pi\left(\frac{r}{Q}\right)$.

Then $\omega^{yrj} = \beta^j$ corresponds to angle $j\theta$ with

$$|j\theta| \leq |J\theta| \leq \pi.$$

So, the terms of $\sum_{j=0}^{J-1} \omega^{yrj} = \sum_{j=0}^{J-1} \beta^j$ span only angle π.



Simple calculation: $\frac{1}{2}$ the terms make angle $\leq \frac{\pi}{4}$ to resultant vector. Since overall span $\leq \pi$, no vector contributes negatively to resultant vector.

$$\cos\frac{\pi}{4} = \frac{1}{\sqrt{2}}.$$

So, length of resultant vector of $\frac{1}{\sqrt{QJ}}\,\omega^{\gamma \ell}\sum_{j=0}^{J-1}\omega^{\gamma r j}$

$$\geq \underbrace{\frac{J}{2}}_{\text{half terms}}\cdot\underbrace{\frac{1}{\sqrt{2}}}_{\text{contribution per term}}\cdot\frac{1}{\sqrt{QJ}} = \frac{1}{2^{3/2}}\cdot\frac{1}{\sqrt{\frac{Q}{J}}} \geq \frac{1}{4\sqrt{r}}.$$

Conclusion: If $-\frac{r}{2} \leq \gamma r \bmod Q \leq \frac{r}{2}$, then the resulting vector $|y\rangle$ has magnitude $\geq \frac{1}{4\sqrt{r}}$.

We next show many such vectors y exist.

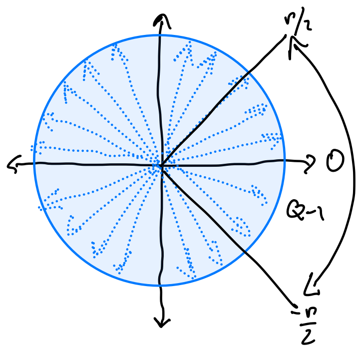
(since $r \in \mathbb{Z}_Q^*$)

If $\gcd(r, Q) = 1$, then $\exists\, r^{-1}$ s.t. $r \cdot r^{-1} \equiv 1 \mod Q$.

Therefore the map $y \longmapsto yr$ is a permutation of $\{0 \dots Q-1\}$.

So, at least r vectors y exist. s.t. $-\frac{r}{2} \leq yr \mod Q \leq \frac{r}{2}$.

Cartoon:



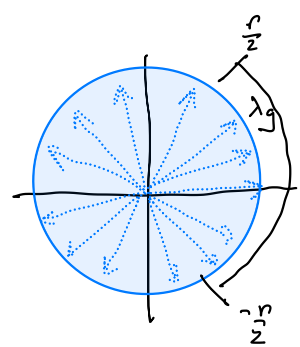
all possible values occur as yr.

So r vectors in region.

When $g := \gcd(r, Q) > 1$, note $g \leq r/2$. Then yr are uniformly distributed over

$$0,\ g,\ 2g,\ \dots,\ \left(\frac{Q}{g} - 1\right) g$$

with $yr = \lambda g$ for g values of y.



If $|\lambda g| < \frac{r}{2} \Rightarrow |\lambda| \leq \frac{r}{2g}$

Then, for at least $2\left\lfloor \frac{r}{2g} \right\rfloor \cdot g \geq \frac{r}{2}$ vectors y,

$$-\frac{r}{2} \leq yr \bmod Q \leq \frac{r}{2}.$$

So, total probability mass on y s.t. $-\frac{r}{2} \leq yr \bmod Q \leq \frac{r}{2}$.

is $\geq \frac{r}{2} \cdot \left(\frac{1}{4\sqrt{r}}\right)^2 = \frac{1}{32}$.

Therefore, we sample a y according to the algorithm for order finding, we correctly calculate ord(x) with probability $\Omega\left(\frac{1}{\log N}\right)$.

So our overall algorithm for factoring is efficient in that it runs in time polylog(N).

Next time: efficient classical algorithms for simulating quantum computation.