

Lecture 20 - Quantum Factoring pt. 2

Say you're trying to factor F . [with hundreds of digits]
[Suffices to find one nontrivial factor - if you want to fully factorize, just recurse.]

Pick some random "multiplier" M ; e.g., 2 [2 might not always work, but often does; we'll come back to this]

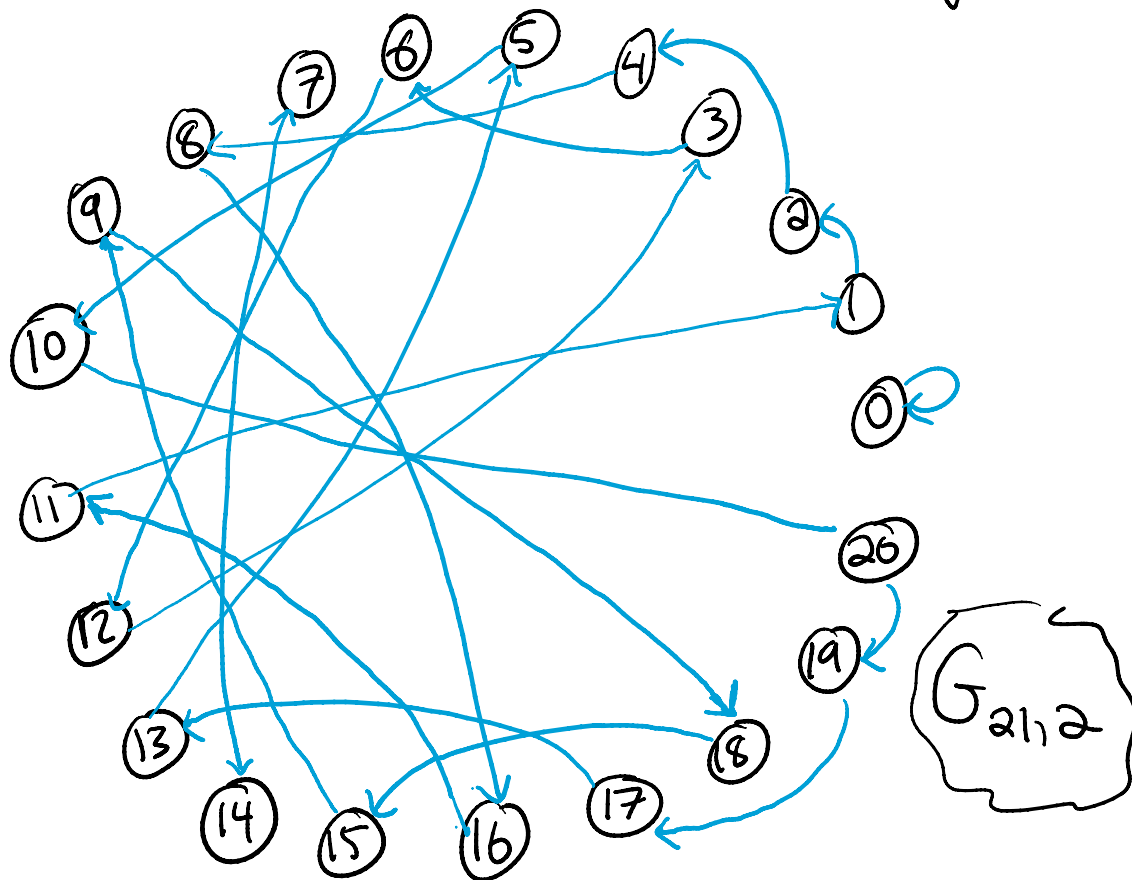
Imagine graph $G_{F,M}$:

vertices : ints mod F

edges : $\text{Next}(x) = M \cdot x \bmod F$

[Obvious : every vertex has] outdegree 1

[Less obvious : does every vertex have] indegree 1?



[[Does vtx ① have indegree 1?]]

How to find W s.t. $2 \cdot W = 1 \pmod F$?

e.g. $F=21$: compute $21 \div 2 = 10.5 \xrightarrow{\text{round}} 11$.

$$\therefore 2 \cdot 11 = 22 = 1 \pmod F.$$

[[If F even.... you've factored F !!]]

So $\text{Prev}(x) = 11 \cdot x \pmod{21}$, since

$$\text{Next}(11 \cdot x) = 2 \cdot 11 \cdot x = 22x = 1x = x \pmod{21}.$$

What if $F = 2183$, $M = 100$?

[[You may recall that one can use Euclid's alg. to find...]]

$$W \text{ s.t. } 100 \cdot W = 1 \pmod{2183}$$

[[but I'll show a slightly alternate way...]]

$$2183 \div 100 = 21.8, \text{ so } 22 \text{ is "close": } 22 \cdot 100 = 2200 = 17 \pmod{2183}.$$

$$2183 \div 17 = 128.41... \text{ so } 128 \cdot 17 \approx 2183$$

$$= 2176 = -7 \pmod{2183}.$$

$$\Rightarrow (128) \cdot (22 \cdot 100) = -7 \pmod{2183}.$$

$$2183 \div (-7) = -311.85... \text{ so } (-312) \cdot (-7) \approx 2183$$

$$= 2184 = 1 \pmod{2183}.$$

$$\Rightarrow (-312) \cdot (128) \cdot (22 \cdot 100) = 1 \pmod{2183}.$$

$$\therefore W := (-312) \cdot (128) \cdot (22) \text{ has } 100 \cdot W = 1 \pmod{2183}!$$

Note: Each remainder is $< \frac{1}{2}$ the previous one
 (100, 17, -7, 1) [in absolute value]

so process takes $\leq \log_2(10^d) = O(d)$ stages
 for d -digit #'s.

And if some remainder is 0.... you found
 a factor of F ! [And you're done!]

So for any M [of our choice, even w/ hundreds
 of digits]

can get W s.t. $\text{Prev}(x) = W \cdot x \bmod F$

[or else we get
 a factor of F]

$\therefore$ indegrees are 1,

vtx ① is on a cycle of (mystery) length L .

[Last time we assumed we could make the following q. op:]

$U: |x\rangle \mapsto |\text{Next}(x)\rangle$. [Can we? We only know how to
 make "Add $\text{Next}(x)$ to Ans".]

[Solution:] $U(X_1, \dots, X_n)$:

$\overbrace{X_1, \dots, X_n} \quad \overbrace{\text{Ans}_1, \dots, \text{Ans}_n}$

[We can indeed
 do it, using
 ignorable temp
 bits, and our ability
 to quantize Prev
 code too.]

• Make $\text{Ans}_1, \dots, \text{Ans}_n$

: $|x\rangle \otimes |00\dots 0\rangle$

• Add $\text{Next}(X_i)$ to Ans's

: $|x\rangle \otimes |\text{Next}(x)\rangle$

• Swap X 's & Ans's

: $|\text{Next}(x)\rangle \otimes |x\rangle$

• Add $\text{Prev}(X_i)$ to Ans's

: $|\text{Next}(x)\rangle \otimes |00\dots 0\rangle$

Because $x \oplus \text{Prev}(\text{Next}(x)) = 00\dots 0$ ↑

Key: Can also make highly efficient quantum code for U^{2^d} : much faster than repeating U for 2^d times! [This is needed to get efficient Revolver Resolver.]

Equivalent to computing " Next^{2^d} "(x)
 $\hookrightarrow = M^{2^d} \cdot x \bmod F.$

Compute $M^{2^d} \bmod F$ by repeated squaring!

$M \rightarrow M^2 \bmod F \rightarrow M^4 \bmod F \rightarrow M^8 \bmod F \rightarrow \dots$

[If F has n digits, can square-mod- F in $O(n^2)$ (indeed, $\tilde{O}(n)$) steps, hence can compute $M^{2^d} \bmod F$, and $\text{Next}^{2^d}(x)$, in $\tilde{O}(n \cdot d)$ steps. Actually, can get any d -digit exponent without much more effort ("Modular exponentiation", cf. 15-251), but Rev.Res. only needs exponents like 2^d .]

Conclusion: Revolver Resolver for U only takes $\tilde{O}(n \cdot d)$ steps to get " θ " to d digits of precision (not $\approx 2^d$ steps!!)

($n = \# \text{ digits in } F$)

[Back to last lecture...]

Recall: Say vertex ① in $G_{F,M}$ is on cycle of length L . We studied Revolver Resolver on " U ", mathematically found states

$$|SW_0\rangle, |SW_1\rangle, \dots, |SW_{L-1}\rangle \text{ s.t. :}$$

⊕ Actually, I only proved that $|SW_k\rangle \perp |SW_{k'}\rangle$, not that moreover $P_k \perp P_{k'}$. But this is indeed true: exercise

- U' rotates $|SW_k\rangle$ in a 2-d subspace P_k by angle $\frac{1}{2\pi} \cdot \frac{K}{L}$, and P_k 's are orthogonal. ⊕

- For $|start\rangle := \frac{1}{\sqrt{L}} (|SW_0\rangle + |SW_1\rangle + \dots + |SW_{L-1}\rangle)$,

$|start\rangle = |00\dots 01\rangle \otimes |0\rangle$, a state we can easily create.

The Alg.: Plug $|start\rangle$ into Revolver Resolver using $d = 10n$ digits of precision.

Claim: (cf hwk #8.3) Result is as if K picked uniformly at random from $0 \dots L-1$, and you got Revolver Resolver for that $|SW_k\rangle$

Alternative way to achieve claim: [you don't actually have to do this, but if you do, it's easier to see the claim]

Given $|\text{start}\rangle$ on qubits $A_1, \dots, A_n$, make qubits $B_1, \dots, B_n$, do "Add A_i to B_i " $\forall i$. Now state is

$$\sqrt{\frac{1}{L}} |SW_0\rangle_{A's} \otimes |SW_0\rangle_{B's} + \sqrt{\frac{1}{L}} |SW_1\rangle_{A's} \otimes |SW_1\rangle_{B's} + \sqrt{\frac{1}{L}} |SW_2\rangle_{A's} \otimes |SW_2\rangle_{B's} + \dots$$

Since $|SW_0\rangle, \dots, |SW_{L-1}\rangle$ are orthonormal, in principle smart Bob could take B qubits to his lab and measure in basis containing them.

Then w/prob $\frac{1}{L}$, Bob sees " SW_0 ", state collapses to $|SW_0\rangle_{A's} \otimes |SW_0\rangle_{B's}$
" " " " " SW_1 " " " " $|SW_1\rangle_{A's} \otimes |SW_1\rangle_{B's}$
" " " " " SW_2 " " " " $|SW_2\rangle_{A's} \otimes |SW_2\rangle_{B's}$

Now Alice runs Rev.Res. on A qubits.

Her outcome is indeed like doing RevRes on a uniformly random $|SW_k\rangle$.

But Bob was measuring ^{*}alone in his office.
Could have measured after Alice, and it wouldn't change probabilities of what Alice sees.

Indeed, Bob doesn't have to measure at all!

Which is good, b/c we don't know how to do Bob's measurement!

[But note that the Adding is essential here.]

Upshot: We have a quantum black box that, with $\tilde{O}(n^2)$ work, gives $10n$ digits of precision to $\frac{K}{L}$ for a random K . *[[Can use box multiple times, but will be a different random K each time.]]*

[[There is no more quantum now. Rest is "elementary" number theory!]]

Later today: Can use box to get L , efficiently.

Now: By getting L , can factor F .

[[This trick was known for decades, prior to quantum computing existing.]]

We'll assume for simplicity $F = P \cdot Q$ for primes P, Q .

[[This is the RSA case, and also the "hardest" case anyway. The more prime factors F has, the smaller they are, and the easier to find. It's possible to handle the general F case with, like, 10% more number theory.]]

Given F , we pick some " M ". Then we get

L , the least val. such that $M^L \equiv 1 \pmod{F}$.

Now compute integer $L/2$. If L was odd $\rightarrow$ M was "unlucky"!

Now compute $S := M^{L/2} \bmod F$.

Note: $S^2 = (M^{L/2})^2 = M^L = 1 \bmod F$.

If $S = -1 \bmod F \rightarrow M$ was "unlucky"!

Else $S+1 \not\equiv 0 \bmod F$; i.e., $S+1$ not a multiple of $P \cdot Q$.

Also, $S-1$ not a multiple of $P \cdot Q$, else

$\bmod F$: $S-1=0 \Rightarrow S=1 \Rightarrow M^{L/2}=1 \Rightarrow$ first power of M to be 1 is $L/2$

But $S^2=1 \Rightarrow S^2-1=0 \Rightarrow (S-1)(S+1)=0 \bmod F$;

i.e., $(S-1)(S+1)$ is a multiple of $P \cdot Q$.

$\therefore P, Q$ in prime factorization of $(S-1)(S+1)$.

Can't have P, Q both in $S-1$ or both in $S+1$, since both not multiples of $P \cdot Q$.

$\therefore P$ divides one of $S-1, S+1$, Q divides other.

$\therefore \gcd(S-1, F) = P$, $\gcd(S+1, F) = Q$ or vice versa

known
efficiently computable.

known



[[What about unluckiness??]]

def: Multiplier M is "lucky" if L is even,
 $M^{L/2} \not\equiv -1 \pmod{F}$.

heuristic fact: $M=2,3$ "almost always" lucky

elementary & theory fact: [[won't prove, will be on homework]]

$\Pr [\text{random } M \leq F \text{ is lucky}] \geq \frac{1}{2}$.

[[Only as low as $\frac{1}{2}$ for very freakish F .
For most F , it's much higher.]]

So can find a lucky M [[and thereby factor F]]
with high proba by picking a few at random.

Final quirky problem: ~~[[Forget everything prior, this is a one-off problem.]]~~

Let L be an unknown n -digit integer.

k is chosen at random [[secretly]] and you get

$\frac{k}{L}$ to 10 n decimal digits. Can you determine L ?

[[Well, no. Imagine L is 100, k chosen to be

44, so $\frac{k}{L} = \frac{44}{100} = .44$. You can notice $\frac{k}{L} = \frac{11}{25}$, but

maybe $k=22, L=50$ or $k=44, L=100$ or $k=132, L=300$ ]]

[[Lazy analyst's way to deal with this issue that the best you can hope for is to find $\frac{K}{L}$ in "lowest terms":]]

fact: # of primes between $\frac{L}{2}$ & L is $\approx \frac{1.6}{n} \cdot L$,
["Weak Prime # if L is n digits.

Theorem", easy to prove]]

$$\therefore \Pr[K > \frac{L}{2} \text{ \& } K \text{ prime}] \approx \frac{1.6}{n}$$

If this happens, $\frac{K}{L}$ is already in lowest terms, so finding it $\Rightarrow$ finding L .
[[only common factor could be K , but $K > L/2$]]

$\frac{1.6}{n}$ is small, but not that small. [[Think $n \approx$ few hundreds]]

Could repeat Black Box $\approx n$ times until we get a prime $K > L/2$.

[[Actually, this is overkill. It's not hard to show that if you just do the Black Box twice, determine $\frac{K_1}{L}$, $\frac{K_2}{L}$ in lowest terms, take L.C.M. of denoms, it gives L with high probability.]]

Final final problem: Given $\frac{K}{L}$ to 10 n digits of accuracy, find $\frac{K}{L}$ in lowest terms

This is a fun problem!

I'll illustrate its solution by demo.

Note: Why should we believe 10^n digits of precision suffices to determine $\frac{K}{L}$?

Well, how close are the closest two fractions with n -digit denoms?

$$\frac{K_1}{L_1} - \frac{K_2}{L_2} = \frac{K_1 L_2 - K_2 L_1}{L_1 L_2} \leftarrow \geq 1 \text{ assuming } \frac{K_1}{L_1} \neq \frac{K_2}{L_2}$$
$$\leftarrow \leq 2^n\text{-digit \#}$$

So they're at least 10^{-2^n} apart.

So even $2n+1$, or $3n$ **[[let alone $10n$]]** digits of accuracy should be enough.