Quantum Algorithms

DPV Chapter 10

Jim Royer

CIS 675

April 24, 2019

Uncredited diagrams are from DPV or homemade.

Shor's Algorithm for Factoring: Background

- In 1994, Peter Shor came up with $O(n^3)$-time algorithm for factoring n-bit integers — on a quantum algorithm.
- Why is this a big deal?
  - $O(e^{n^{1/3}(\log n)^{2/3}})$-time is the best known runtime for factoring on "nonquantum" computers.
  - The security of RSA and many other cryptosystems depend on the hardness of factoring.
  - Factoring was one of the first "natural" problems on which quantum computation appeared to have a real advantage.
    (There are more such problems now, but still not that many.)
- Chapter 10 of DPV sketches Shor's Algorithm.
- Here we will sketch Chapter 10 of DPV.

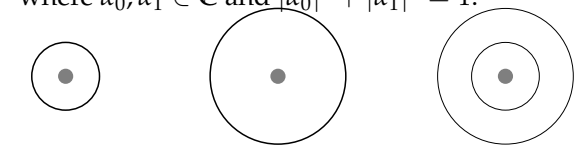
Representing Bits

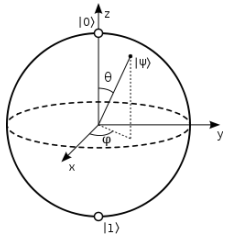
- The standard way to represent a bit is with voltages on a wire:
  - a low voltage for **0**.
  - a high voltage for **1**.
- An alternative is to use a single electron and use the electron's state:
  - its ground state represents **0**.
  - its excited state to represent **1**.
- An electron can be in a mixture (superposition) of its ground and excited states.
- We can used these superposition to achieve a kind of parallelism ...

Qubits & Superposition

An electron of a hydrogen atom can be in:

- a ground (low energy) state, denoted $|0\rangle$, or
- an excited (high energy) state, denoted $|1\rangle$, or
- $\alpha_0|0\rangle + \alpha_1|1\rangle$, a linear combination of $|0\rangle$ and $|1\rangle$ where $\alpha_0, \alpha_1 \in \mathbb{C}$ and $|\alpha_0|^2 + |\alpha_1|^2 = 1$.





Superposition principle

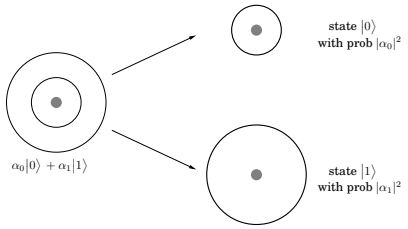
If a quantum system can be in one of two states, s_0 and s_1, then it can also be in any linear superposition of s_0 and s_1.

There are many proposed physical representations for qubits.

Qubits & Measurement

When we measure a qubit $(\alpha_0|0\rangle + \alpha_1|1\rangle)$, we force it into:

- state $|0\rangle$ with probability $|\alpha_0|^2$, or else
- state $|1\rangle$ with probability $|\alpha_1|^2 = 1 - |\alpha_0|^2$.



Quantum Registers, 1

- Quantum register ≡ an array of qubits
- For a two qubit register:

$$\begin{aligned}&(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle)\\ &\quad = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle\end{aligned}$$

  - Note that $(\alpha_0\beta_0)^2 + (\alpha_0\beta_1)^2 + (\alpha_1\beta_0)^2 + (\alpha_1\beta_1)^2$
    $= \alpha_0^2 \cdot (\beta_0^2 + \beta_1^2) + \alpha_1^2 \cdot (\beta_0^2 + \beta_1^2)$
    $= \alpha_0^2 \cdot 1 + \alpha_1^2 \cdot 1 = 1.$
  - The probability of obtaining $|ij\rangle$ from a measurement is: $|\alpha_i\beta_j|^2$.
  - If we measure just the first qubit and obtain $|0\rangle$, the state of the register becomes

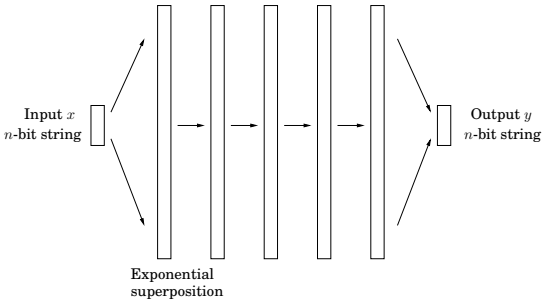
$$\frac{\alpha_0\beta_0}{r}|00\rangle + \frac{\alpha_0\beta_1}{r}|01\rangle, \text{ where } r = \sqrt{|\alpha_0\beta_0|^2 + |\alpha_0\beta_1|^2}.$$

Quantum Registers, 2

- Quantum register ≡ an array of qubits
- For an n qubit register:
  - Can achieve the superimposition of all 2^n classical states.
  - If you put this through a series of reversible circuits, no states are collapsed.

$\therefore$ You have 2^n many classical computations going on at the same time.

!!! The problem is learning anything useful via measurements.



The Plan for Factoring in Quantum Poly-Time

1. FACTORING is reduced to finding a nontrivial square root of 1 modulo N.
2. Finding such a root is reduced to computing the order of a random integer modulo N.
3. The order of an integer is precisely the period of a particular periodic superposition.
4. Finally, periods of superpositions can be found by the quantum FFT.

The Discrete Fourier Transform, 1

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{M-1} \end{bmatrix} = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & & \vdots & & \\ 1 & \omega^j & \omega^{2j} & \dots & \omega^{j(n-1)} \\ \vdots & & \vdots & & \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}$$

- $\vec{\beta}, \vec{\alpha} \in \mathbb{C}^M$,
- ω = a complex M^{th} root of unity,
- $\frac{1}{\sqrt{M}}$ is a fudge so that $\vec{\alpha} \cdot \vec{\alpha}^* = 1$.
- The classical *Fast Fourier Transform* algorithm computes this transform in $\Theta(M \log M)$ time.

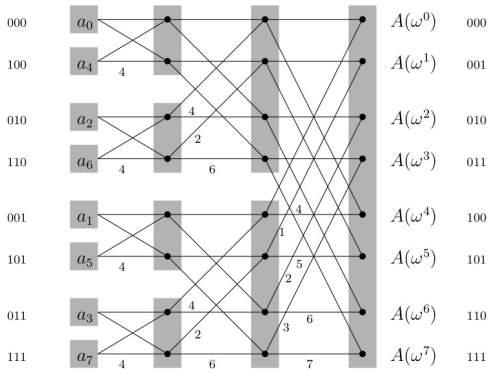
The Discrete Fourier Transform, 2

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{M-1} \end{bmatrix} = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & & \vdots & & \\ 1 & \omega^j & \omega^{2j} & \dots & \omega^{j(n-1)} \\ \vdots & & \vdots & & \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}$$

- The quantum version of FFT takes $O((\log M)^2)$ time!!!.
- Input/Output $(\log_2 M) = m$ qubits in superimposition
- $\vec{\alpha} = \sum_{j=0}^{M-1} \alpha_j |j\rangle$, where $|j\rangle = |j_{m-1} \dots j_1 j_0\rangle$
  and $(j_{m-1} \dots j_1 j_0)_2$ = the binary rep of j.

The Quantum Fast Fourier Transform, 1

The Classical circuit for FFT



- There is a reversible version of this that takes m stages with $O(m)$ operations per stage.
- Hence the $O(m^2) = O((\log M)^2)$ run time.

The Quantum Fast Fourier Transform, 2

However:

- The output comes from a measurement.
- The output will be $|j\rangle$ (where j is an m bit vector) with probability $|\beta_j|^2$ (for $j \in \{0, \dots, M-1\}$).
- *So this is more like sampling than straightforward computation.*

QFT

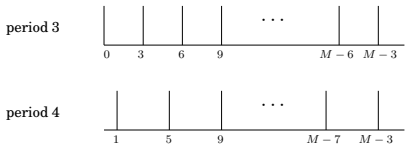
Input: A superposition of $m = \log_2 M$ qubits $|\vec{\alpha}\rangle = \sum_{j=0}^{M-1} \alpha_j |j\rangle$.

Method: Using $O(m^2)$ quantum/reversible operations perform the quantum version of FFT to obtain the superposition $|\vec{\beta}\rangle = \sum_{j=0}^{M-1} \beta_j |j\rangle$.

Output: A random m-bit number $j \in \{0, \dots, M-1\}$ from the probability distribution $Prob[j] = |\beta_j|^2$.

Periodicity, 1

- $|\vec{\alpha}\rangle = (\alpha_0, \alpha_1, \ldots, \alpha_{M-1})$ is periodic with period k and offset j $\iff$
  - $\alpha_j = \alpha_{j+k} = \alpha_{j+2k} = \cdots = \alpha_{j-k+M} = a > 0$,
  - all the other α_i's is 0,
  - k divides M and $0 \leq j < k$.



Fact

Suppose $|\vec{\alpha}\rangle$, the input to QFT, is periodic with period k and offset 0.
Then $|\vec{\beta}\rangle$, the QFT output, is periodic with period M/k and offset 0; and
when we measure $|\vec{\beta}\rangle$, we get one of $0, \frac{M}{k}, \frac{2M}{k}, \ldots, \frac{(k-1)M}{k}$ with each probability $\frac{k}{M}$.

Periodicity, 2

Fact

Suppose $|\vec{\alpha}\rangle$, the input to QFT, is periodic with period k and offset 0.
Then $|\vec{\beta}\rangle$, the QFT output, is periodic with period M/k and offset 0; and
when we measure $|\vec{\beta}\rangle$, we get one of $0, \frac{M}{k}, \frac{2M}{k}, \ldots, \frac{(k-1)M}{k}$ with each probability $\frac{k}{M}$.

Lemma

Suppose s independent samples are drawn uniformly from

$$0, \frac{M}{k}, \frac{2M}{k}, \ldots, \frac{(k-1)M}{k}$$

Then with probability $\geq 1 - \frac{k}{2^s}$, the GCD of these samples is $\frac{M}{k}$.

To see why this is useful, we need to bring a few more pieces in to play.

Factoring as Periodicity, 1

Suppose N is a positive integer.

- A nontrivial factor of N is a $k \in \{2, \ldots, N-1\}$ that divides N.
  I.e., 3 and 5 are nontrivial factors of 15.
- A nontrivial square root of N is an integer x such that
  $x \not\equiv \pm 1 \pmod{N}$ and $x^2 \equiv 1 \pmod{N}$.
  I.e., 4 is a nontrivial square of 15
  since $4 \not\equiv \pm 1 \pmod{15}$ *and* $4^2 = 16 \equiv 1 \pmod{15}$.

Lemma

Suppose x is nontrivial square root of N.
Then both $\gcd(x-1, N)$ and $\gcd(x+1, N)$ are nontrivial factors of N.

Proof.

$x^2 \equiv 1 \pmod{N} \iff x^2 - 1 = a \cdot N \iff N$ divides $x^2 - 1$.
But $x^2 - 1 = (x-1)(x+1)$ and $x \not\equiv \pm 1 \pmod{N}$.
So N fails to divide both $x-1$ and $x+1$.
Therefore, $1 < \gcd(x-1, N), \gcd(x+1, N) < N$. □

Factoring as Periodicity, 2

Suppose N is a positive integer.

- The order of x modulo N is the least integer $r > 0$ such that $x^r \equiv 1 \pmod{N}$.
  I.e., The order of 2 modulo 15 is 4, since:

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$2^r \bmod 15$	2	4	8	1	2	4	8	1	2	4	8	1	2	4

Lemma

Suppose

- *N is an odd number that at least two different primes divide.*
- *x is chosen uniformly at random from $\{0, \ldots, N-1\}$.*
- *$\gcd(x, N) = 1$.*
- *r is the order of x modulo N.*

Then with probability $\geq \frac{1}{2}$, r is even and $x^{r/2}$ is a nontrivial square root of 1 mod N.

Example

The order of 2 modulo 15 is 4.
So $2^2 = 4$ is a nontrivial root of 1 modulo 15.
So $\gcd(4+1, 15) = 5$ is a divisor of 15.

Factoring as Periodicity, 3

- Fix N and x and consider $f(a) = x^a \bmod N$. Let r be the order of x.
- Then $f(0) = f(r) = f(2r) = \cdots = 1, f(1) = f(r+1) = f(2r+1) = \cdots = x$, etc.
- f is periodic with period r.
- f can be computed efficiently via repeated squaring.
- Goal: Use f to set up a periodic superposition with period r.
  *(See the **Setting up a periodic superposition** box in the text.)*
- Then QFT can find r.

Shor's Algorithm

Input: an odd composite integer N.

Output: a factor of N.

1. Choose x uniformly at random in the range $1 \le x \le N-1$.
2. Let M be a power of 2 near N (for reasons we cannot get into here, it is best to choose $M \approx N^2$).
3. Repeat $s = 2 \log N$ times:
   (a) Start with two quantum registers, both initially 0, the first large enough to store a number modulo M and the second modulo N.
   (b) Use the periodic function $f(a) \equiv x^a \bmod N$ to create a periodic superposition $|\alpha\rangle$ of length M as follows (see box for details):
      i. Apply the QFT to the first register to obtain the superposition $\sum_{a=0}^{M-1} \frac{1}{\sqrt{M}} |a, 0\rangle$.
      ii. Compute $f(a) = x^a \bmod N$ using a quantum circuit, to get the superposition $\sum_{a=0}^{M-1} \frac{1}{\sqrt{M}} |a, x^a \bmod N\rangle$.
      iii. Measure the second register. Now the first register contains the periodic superposition $|\alpha\rangle = \sum_{j=0}^{M/r-1} \sqrt{\frac{r}{M}} |jr + k\rangle$ where k is a random offset between 0 and $r-1$ (recall that r is the order of x modulo N).
   (c) Fourier sample the superposition $|\alpha\rangle$ to obtain an index between 0 and $M-1$.

   Let g be the gcd of the resulting indices $j_1, \ldots, j_s$.
4. If M/g is even, then compute $\gcd(N, x^{M/2g} + 1)$ and output it if it is a nontrivial factor of N; otherwise return to step 1.

From the previous lemma, this works for about $\frac{1}{2}$ the choices of x.

Another Application of QFT: Discrete Log

- Let $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ where p is a prime.
- $g \in \mathbb{Z}_p$ is a *generator* when $\{g^i \bmod p : 1 \le i \le p-1\} = \{1, 2, \ldots, p-1\}$.
- I.e., $(\forall x \in \{1, \ldots, p-1\})(\exists \ell_x \in \{1, \ldots, p-1\})[x = g^{\ell_x} \bmod p]$,
  ℓ_x as above is called the *discrete log of x with basis g*.
- Finding discrete logs is hard classically.
- This hardness is the basis of several cryptosystems.
- You can use the QFT to solve discrete log problems in poly-time.

- So quantum computing may (eventually) force big changes in cryptography and security.