Quantum Algorithms (pt. 2)

CSE 490Q: Quantum Computation
• D. Deutsch (1985) showed that quantum computers could solve a problem more efficiently than any classical algorithm
  • first hints of the extra algorithmic power available to QCs
  • work was later extended, jointly with R. Josza

• Exponential reduction in the number of queries (from $\Theta(2^n)$ to 1)

• Limitations of the result
  1. Bound is on queries not running time
  2. No practical application of this ("toy" problem)
  3. Assumes a black box, which is not allowed in BPP
  4. Compares to P rather than BPP
Shor’s Algorithm

• Shor’s (1994) breakthrough result was an efficient quantum algorithm for factoring
  • strongly believed to be classically hard
    • 200+ years of failed attempts to solve it
    • hardness is assumed by cryptosystems like RSA
  • not a toy problem, not a black box

• Fastest classical algorithm runs in $O(2^{n/3})$ time
• Shor’s algorithm is $O(n^3)$
Factoring by Order Finding

• Shor’s algorithm actually solves a different problem called “order finding”
  • Problem: given a number \( y < N \), find the smallest \( r \) such that \( y^r = 1 \) (mod \( N \))

• Previously known that this was sufficient to solve factoring
  • (won’t need the details of that reduction again)

• “Order finding” is easily generalized into “period finding”
  • Problem: given a function \( f \), with \( n \) input and output bits, find the smallest \( x \) such that \( f(r+x) = f(r) \) for all \( r \)
Post-Shor Algorithms

• Much later work focused on *generalizing* it
  • we will look at some of the generalizations (more than one!)
  • Shor’s paper still has insights that may not be fully understood still

• First generalization was the *phase estimation algorithm*
Alexei Kitaev, in Russia at the time, heard about Shor’s result. Unable to get the paper, he re-derived how he thought it must work. His result was actually a significant generalization of Shor’s.

The technique he invented is called “phase estimation”.

(He also generalized the problem solved.)
Phase Estimation

**Problem**: Given a state $|\psi\rangle$ that is an eigenvector of an n-qubit unitary $V$ corresponding to eigenvalue $\exp(2\pi i \phi/2^n)$, for some integer $\phi$, find the value of $\phi$

- Any eigenvalue $z$ of $V$ satisfies $|z| = 1$, so it is $\exp(2\pi i \Theta)$ for some $0 \leq \Theta < 1$

- We are assuming $\Theta$ has $n$ digits after the decimal point (written in binary)
  - in practice, we will find an integer close to $2^n \Theta$
Quantum Fourier Transform

• Key ingredient in phase estimation is a Quantum Fourier Transform (QFT):

\[ |j\rangle := \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k j / 2^n} |k\rangle \]

• Basis of binary strings \(0^n, \ldots, 1^n\) can be identified with binary numbers
  • e.g., \(000 = 0, 001 = 1, 010 = 2, 011 = 3, \ldots, 111 = 7\)
  • in general, string \(1^n\) corresponds to number \(2^n - 1\)

• We will make this identification frequently going forward
Quantum Fourier Transform

• QFT is a unitary:

\[
\| F |1\rangle |1\rangle \|^2 = \left\| \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k/2^n} |k\rangle |1\rangle \right\|^2
\]

\[
= \sum_{k=0}^{2^n-1} \frac{1}{2^n} = 2^n \cdot \frac{1}{2^n} = 1
\]

• Not obvious that this unitary can be implemented, but it can be
  • we will sketch this out in the homework
  • (closely related to the classical FFT algorithms)
Phase Estimation Circuit
Phase Estimation Analysis 1

\[ A \]
\[ \left( \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{C}_0}^n |x \rangle \otimes |v \rangle \right) \otimes |v \rangle \]

\[ B_0 \]
\[ [C U]_0 \left( \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{C}_0}^n |x \rangle \otimes |v \rangle \right) \]
\[ = [C U]_0 \left( \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{C}_0}^n |x \rangle \otimes (|0 \rangle + i|x \rangle) \otimes |v \rangle \right) \]
\[ = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{C}_0}^n |x \rangle \otimes (|0 \rangle \otimes |v \rangle + |1 \rangle \otimes |v \rangle) \]
\[ = \frac{1}{\sqrt{2^n}} \sum_{x = 0}^{2^n-1} (|x \rangle \otimes _\mathbb{C}_0 |0 \rangle + e^{2\pi i x \varphi} |1 \rangle \otimes |v \rangle) \]
Phase Estimation Analysis 2

\[ \left[ CV^2 \right]_1 \left( \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \langle x|_N + e^{2\pi i (\alpha/2^n)} |1\rangle \langle 1|_N \right) \]

\[ = \left[ CV^2 \right]_1 \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-2} \left( |x\rangle \langle x|_N + |1\rangle \langle 1|_N \right) + e^{2\pi i (\alpha/2^n)} \left( |0\rangle \langle 0|_N + |1\rangle \langle 1|_N \right) \]

\[ = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-2} \left( |x\rangle \langle x|_N + e^{2\pi i (\alpha/2^n)} |x\rangle \langle x|_N \right) + e^{2\pi i (\alpha/2^n)} \left( |0\rangle \langle 0|_N + e^{2\pi i (\alpha/2^n)} |1\rangle \langle 1|_N \right) \]

\[ = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-2} \sum_{k=0}^{3} e^{2\pi i kx/2^n} |x\rangle \langle x|_N \]

\[ + e^{2\pi i (\alpha/2^n)} \left( |0\rangle \langle 0|_N + e^{2\pi i (\alpha/2^n)} |1\rangle \langle 1|_N \right) \]
\[ \left( \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i k x / 2^n} |x, 0, k, u \rangle \right) \]

\[ = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i k x / 2^n} \left( |x, 0, k, v \rangle + e^{-\pi i 2^k / 2^n} |x, 1, k, v \rangle \right) \]

\[ = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \sum_{k=0}^{2^n-1} \sum_{l=0}^{2^n-1} e^{2\pi i k x / 2^n} \left( |x, l, k, v \rangle \right) \]

\[ r = n - 1 \quad h = (n - 1) - 1 = 0 \]
\[ 
\begin{align*}
\mathbb{E}^+ (c) &= f^+ \left( \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k \phi / 2^n} |k\rangle |\phi \rangle |1\rangle |u\rangle \right) \\
&= |\psi \rangle \otimes |u\rangle
\end{align*}
\]
Phase Estimation Analysis 5
Phase Estimation

- Measuring the final state gives $|\phi>$ with certainty
  - (in reality, there will be errors, but it is ok to fail 1/3 of the time)

- Most expensive part of the algorithm is the inverse QFT
  - can bound this by $O(n^2)$ gates

- Algorithm has many uses (see future HW), but we care most about...
Problem: given a number $y < N$, find the smallest $r$ such that $y^r = 1 \pmod{N}$

Solution:

• Choose $n \geq \log_2 N$ so we can write 0, ..., $N-1$ in $n$ bits
• Choose $V$ to be the unitary taking $|x>$ to $|xy \mod N>$
  • make classical multiplication and mod into reversible circuits
  • this is unitary: inverse takes $|x>$ to $|xy^{-1} \mod N>$
    • ($y^{-1}$ exists provided $\gcd(y, N) = 1$)
• What is $|v>$?
Order Finding By PE

- Eigenvectors of $V$:

\[
\Psi_s > = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i ks/r} |y_k >_{y \mod N} > s \quad s=0, \ldots, r-1
\]

\[
\Psi'_{s>} = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{2\pi i ks/r} |y_k y' \mod N >
\]

\[
= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{2\pi i ks/r} |y_k >_{y' \mod N}
\]

\[
= e^{2\pi i s/r} \Psi_s >
\]
• Phase estimation on $|v_s\rangle$ would return $2^n s / r$ with certainty
  • could then find $r = 2^n s / (2^n s / r)$

• But there’s a problem:
  • we can’t prepare $|v_s\rangle$ because we don’t know $r$
  • (definition of $|v_s\rangle$ uses $r$ repeatedly)

• Need to be more clever...
Order Finding By PE

- Cannot prepare $|v_s\rangle$ but can prepare

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |0_s\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \sum_{k=0}^{r-1} e^{2\pi i k s/r} |y_k \text{ mod } N\rangle$$

$$= \sum_{k=0}^{r-1} \left( \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{2\pi i k s/r} \right) |y_k \text{ mod } N\rangle$$

$$= e^{2\pi i k / \sqrt{r}} |y_k \text{ mod } N\rangle$$

$$\sum_{k=0}^{r-1} e^{2\pi i k} = \left( e^{2\pi i} \right)^{r} = 1$$

$$= \frac{1}{r} \left( \cos \frac{\pi}{r} \sum_{j=0}^{r-1} e^{2\pi i j} + \sin \frac{\pi}{r} \sum_{j=0}^{r-1} \sin 2\pi j \right) = \frac{1}{r} \frac{r^5 - 1}{r - 1} = 0$$

$$= |y^0 \text{ mod } N\rangle = \langle 1 |$$
Order Finding By PE

- Can prepare $|1\rangle = \text{uniform superposition of } |v_s\rangle$, for $s = 0, 1, \ldots, r-1$

- For $|v_s\rangle$, the algorithm returns $2^n s / r$ with certainty
  - dividing the returned value by $2^n$ gives the fraction $s / r$

- If we do all of this and then measure the resulting state, we get $s / r$ for a random choice of $s$ in $0, 1, \ldots, r-1$
  - may not be exact but errors will be on the order of $1/2^n$
• Repeating this multiple times gives estimates of $s / r$ for random choices of $s$

• Classical algorithm, based on the method of continued fractions, will find $s / r$
  • requires the errors are no more than $1/ 2r^2$
  • runs in $O(n^3)$ time