Quantum Algorithms (pt. 2)

CSE 490Q: Quantum Computation

Deutsch-Josza

- 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 q. 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 $y^r = 1 \pmod N$
- 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

Phase Estimation "History"

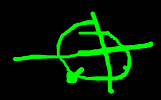
Alexei Kitaev, in Russia at the time, heard about Shor's result. Unable to get the paper, he re-derived how he thought it must work. His result was actually a significant generalization of Shor's.

The technique he invented is called "phase estimation"

(He also generalized the problem solved.)

Phase Estimation

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



$e^{2\pi i \Theta}$

$\frac{s}{r}$

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

- We are assuming Θ 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):

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

$$F^\dagger$$

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

$$x_\ell = x_1 x_0$$

$$= x_\ell 2^\ell + \cdots + 2x_1 + x_0$$

$$2^j x_j$$

- We will make this identification frequently going forward

Quantum Fourier Transform

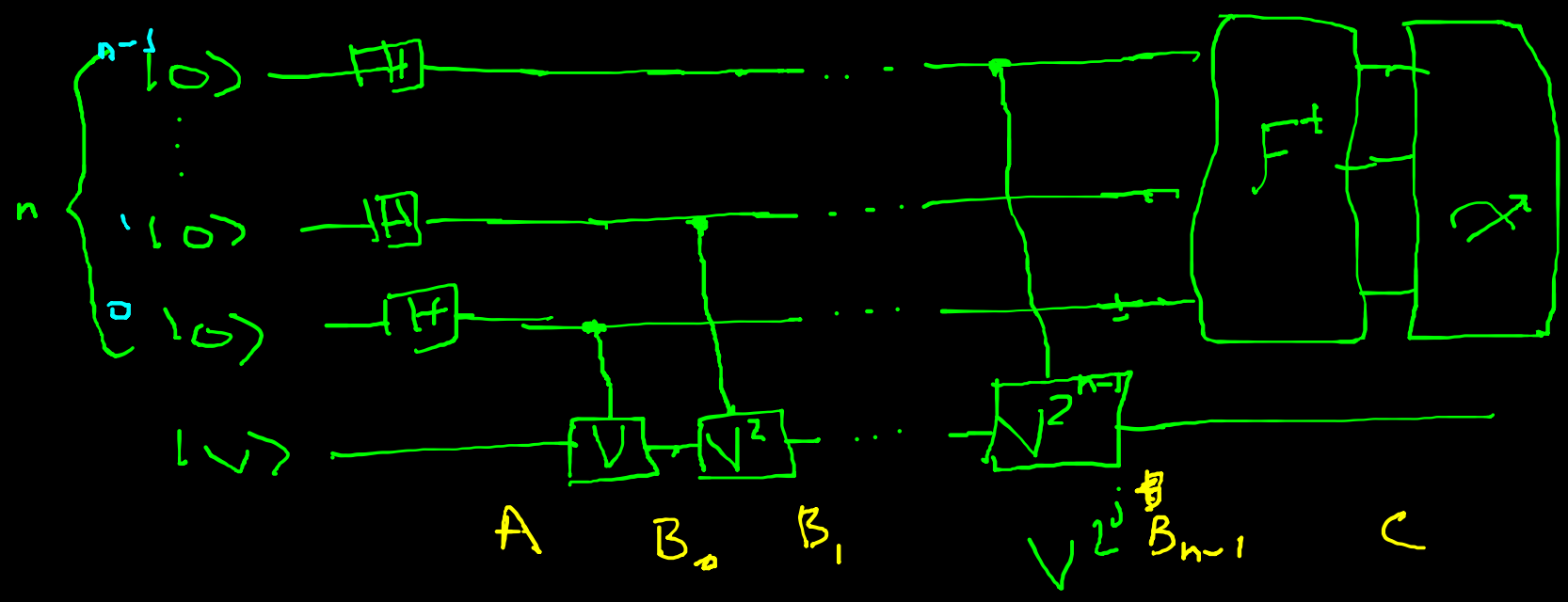
- QFT is a unitary: $F^\dagger F$

$$\| F|j\rangle \|^2 = \left\| \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i\, kj/2^n} |j\rangle \right\|^2$$

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

- Not obvious that this unitary can be implemented efficiently, 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 $\left(\frac{1}{\sqrt{2^n}}\sum_{x\in 0^n}^{1^n} |x\rangle\right)\otimes|v\rangle$

B_0 $[CV]_0\left(\frac{1}{\sqrt{2^n}}\sum_{x\in 0^n}^{1^n} |x\rangle\otimes|v\rangle\right)$

$= [CV]_0\left(\frac{1}{\sqrt{2^n}}\sum_{x=0^{n-1}}^{1^{n-1}} |x0\rangle + |x1\rangle\right)\otimes|v\rangle$

$= \frac{1}{\sqrt{2^n}}\sum_{x\in 0^{n-1}}^{1^{n-1}} |x0\rangle\otimes|v\rangle + |x1\rangle\otimes V|v\rangle$

$= \frac{1}{\sqrt{2^n}}\sum_{x=0^{n-1}}^{1^{n-1}} |x0v\rangle + e^{2\pi i\varphi 1 2^n}|x1,v\rangle$

Phase Estimation Analysis 2

B_1

$$[CV^{2^n}]_1 \left(\frac{1}{\sqrt{2^n}} \sum_{x=0^{n-1}}^{1^{n-1}} |x0\rangle + e^{2\pi i \varphi/2^n} |x1, v\rangle \right)$$

$$= [CV^2]_1 \frac{1}{\sqrt{2^n}} \sum_{x=0^{n-2}}^{1^{n-2}} |x\underline{00}, v\rangle + |x\underline{10}, v\rangle + e^{2\pi i \varphi/2^n} \left(|x\underline{01}, v\rangle + |x\underline{11}, v\rangle \right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0^{n-2}}^{1^{n-2}} |x\underline{00}, v\rangle + e^{2\pi i (2\varphi)/2^n} |x\underline{10}, v\rangle + e^{2\pi i \varphi/2^n} \left(|x\underline{01}, v\rangle + e^{2\pi i (2\varphi)/2^n} |x\underline{11}, v\rangle \right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0^{n-2}}^{1^{n-2}} \sum_{k=0}^{3} e^{2\pi i k \varphi / 2^n} |x, \underline{k}, v\rangle$$

Phase Estimation Analysis 3

$$B_\ell \quad [CV^{2^\ell}]_\ell \left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{1^{n-\ell}} \sum_{k=0}^{2^\ell-1} e^{2\pi i k\varphi/2^n} |x,k,v\rangle \right)$$

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

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

$$k+2^\ell \leq 111\cdots 7 = 2^{\ell+1}-1+2^\ell = 2^{\ell+1}-1$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0^{n-\ell-1}}^{1^{n-\ell-1}} \sum_{k=0}^{2^{\ell+1}-1} e^{2\pi i k\varphi/2^n} |x,k,v\rangle$$

$$\ell = n-1 \qquad n-(n-1)-1 = 0$$

Phase Estimation Analysis 4

$B_{n-1} \longrightarrow \left(\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k\varphi/2^n} |k\rangle \right) \otimes |v\rangle$

$C \quad \underset{(F^{-1} \otimes I)}{F^{-1}} \left(\quad \right) = F^{-1} \left(\frac{1}{\sqrt{2}} \sum_{k=0}^{2^{n-1}} e^{2\pi i k\varphi/2^n} |k\rangle \right) \otimes |v\rangle$

$= \underline{|\varphi\rangle} \otimes |v\rangle$

Phase Estimation Analysis 5

Phase Estimation

φ

- Measuring the final state gives $|\phi\rangle$ 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

$$\frac{\text{poly}(n)}{2^n} \ll \frac{1}{3}$$

- Algorithm has many uses (see future HW), but we care most about…

Order Finding By PE

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 $0, \ldots, N-1$ in n bits
- Choose V to be the unitary taking $|x\rangle$ to $|xy \bmod N\rangle$
  - make classical multiplication and mod into reversible circuits
  - this is unitary: inverse takes $|x\rangle$ to $|xy^{-1} \bmod N\rangle$
    - (y^{-1} exists provided $\gcd(y, N) = 1$)
- What is $|v\rangle$?

$x \mapsto xy \mapsto xyy^{-1}$

$\cdot 1 \mapsto$

x

Order Finding By PE

- Eigenvectors of V:

$$\longrightarrow |v_s\rangle := \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i ks/r} \, |y^k \bmod N\rangle \quad \text{for } s = 0 \ldots r-1$$

$$V|v_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i ks/r} \, |\underbrace{y^k y}_{y^{k+1}} \bmod N\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i (k-1)s/r} \, |y^k \bmod N\rangle$$

$$= \underline{e^{2\pi i \underline{s/r}}} \, |v_s\rangle$$

Order Finding By PE

$$\frac{\varphi}{2^n} = \frac{s}{r}$$

$$\varphi = \frac{2^n s}{r}$$

- 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

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|v_s\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{2\pi i k s/r}\,|y^k \bmod N\rangle$$

$$= \sum_{k=0}^{r-1}\left(\underbrace{\frac{1}{r}\sum_{s=0}^{r-1} e^{2\pi i k s/r}}_{\left(e^{2\pi i k/r}\right)^s}\right)|y^k \bmod N\rangle$$

$$\left[\omega^r = e^{2\pi i k} = \left(e^{2\pi i}\right)^k = 1 \qquad \frac{1}{r}\left(\omega^0 + \omega^1 + \omega^2 \cdots + \omega^{r-1}\right) = \frac{1}{r}\,\frac{\omega^r - 1}{\omega - 1} = 0\right]$$

$$= |y^0 \bmod N\rangle = \boxed{|1\rangle}$$

Order Finding By PE

- Can prepare $|1\rangle$ = uniform superposition of $|v_s\rangle$, for s = 0, 1, …, 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, …, r-1
  - may not be exact but errors will be on the order of $1/2^n$

1/r

2/r

3/r

Order Finding By PE

- 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