Lecture 8, 29 Sep 2020

'93 Bernstein-Vazarani

n bit 1-bit $m=1$

artificial? but unambiguous speed-up

Choose some $a < 2^n$

$$f(x) = a \cdot x = \oplus a_i x_i \quad \text{bitwise XOR}$$

How many invocations of f to determine a?

Classically takes n

Choose $x = 2^m$ $x = (0, \ldots 0, \overset{m^{th}}{1}, 0 \ldots 0)$

then $x \cdot a = a_m$

$m = 0, \ldots, n-1$ so n times to determine each bit of a

"phase kickback"

$$U_f |x\rangle_n \underbrace{\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)} = (-1)^{f(x)} |x\rangle_n \underbrace{\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)}$$

need $H^{\otimes n}|x\rangle$

$$H|x\rangle_1 = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x |1\rangle\right) = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} (-1)^{xy} |y\rangle$$

$$H^{\otimes n}|x\rangle_n = \frac{1}{2^{n/2}} \sum_{y_{n-1}=0}^{1} \cdots \sum_{y_0=0}^{1} (-1)^{\sum_{j=0}^{n-1} x_i y_i} |y_{n-1}\rangle \dots |y_0\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{0 \le y < 2^n} (-1)^{x \cdot y} |y\rangle$$

bitwise dot product mod 2

e.g. $n=2$

$$H^{\otimes 2}|00\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$

$$H^{\otimes 2}|01\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right)$$

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

Will also need this identity

$$\sum_{x_1=0}^{1} \sum_{x_0=0}^{1} (-1)^{(a_0+y_0)x_0+(a_1+y_1)x_1}$$

$$= \left(\sum_{x_1=0}^{1} (-1)^{(a_1+y_1)x_1} \right) \left(\sum_{x_0=0}^{1} (-1)^{(a_0+y_0)x_0} \right)$$

so $$\sum_{x} (-1)^{a\cdot x + y\cdot x} = \prod_{j} \sum_{x_j=0}^{1} (-1)^{(a_j+y_j)x_j}$$

if any $\begin{cases} a_j \neq y_j & (-1)^0 + (-1)^1 = 0 \\ a_j = y_j & (-1)^0 + (-1)^0 = 2 \end{cases}$

so

$$\to 2^n \delta_{a,y}$$

$$\left(H^{\otimes n} \otimes H\right) U_f \left(H^{\otimes n} \otimes H\right) |0\rangle_n |1\rangle$$

$$H^{\otimes n} \otimes H \, U_f \left(\frac{1}{2^{n/2}} \sum_{0 \le x < 2^n} |x\rangle_n \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle\right)\right)$$

$$= \frac{1}{2^{n/2}} \left[H^{\otimes n} \sum_{0 \le x < 2^n} (-1)^{f(x)} |x\rangle |1\rangle \right]$$

$$= \frac{1}{2^n} \left[\sum_x \sum_y (-1)^{a \cdot x + x \cdot y} |y\rangle |1\rangle \right]$$

$$= \frac{1}{2^n} \sum_y 2^n \delta_{a,y} |y\rangle |1\rangle$$

$$= |a\rangle_n |1\rangle !$$

apply U_f once, measure input gives a, such that $f(x) = a \cdot x$

factor of n speedup

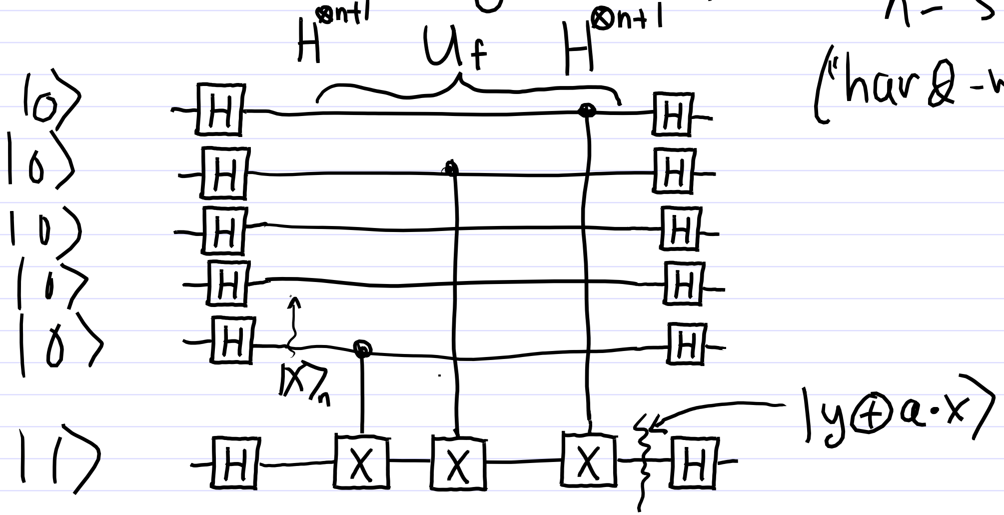
Alternatively (from last time):

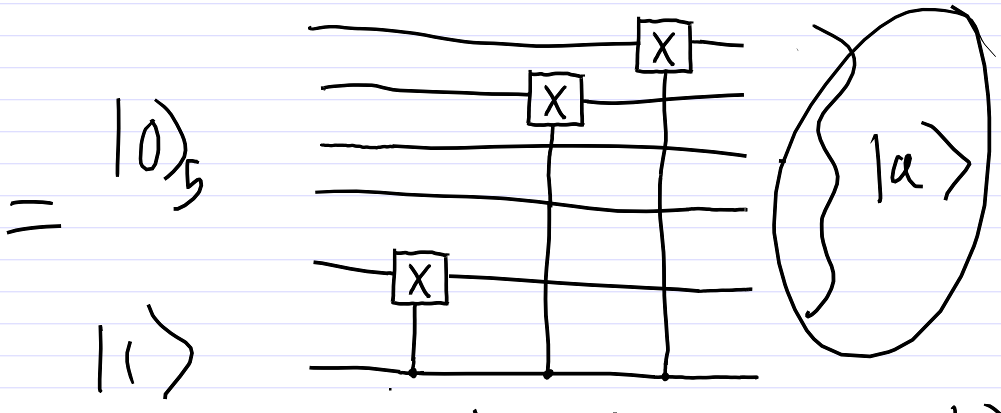
$$U_f |x\rangle_n |y\rangle_1 = |x\rangle_n |y \oplus a\cdot x\rangle_1$$

$a = 11001$

$n = 5$

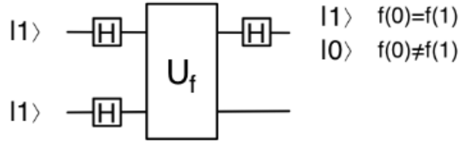
("hard-wired")



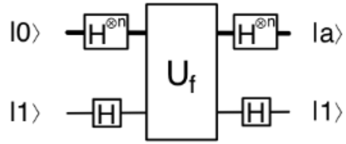


measure input bits, gives $|a\rangle$!

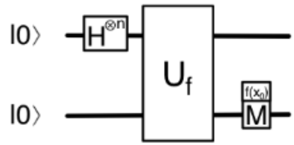
Single invocation of U_f

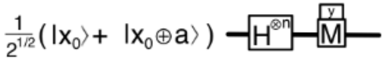


Deutsch '92 (p.44), factor of 2 speedup to determine whether or not 1bit→1bit function $f(x)$ is constant

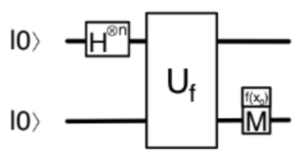


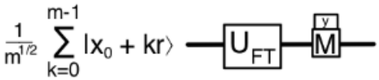
Bernstein–Vazirani '93 (p.52), $f(x) = a \cdot x \equiv \oplus_i a_i x_i$, factor of n speedup to determine a



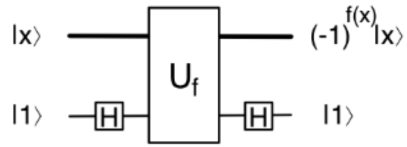


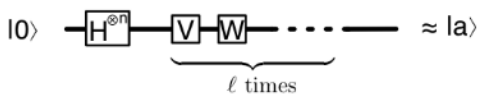
Simon '94 (p.56), $f(x) = f(x \oplus a)$, measured y has $a \cdot y = 0$ (equivalently $\sum_i a_i y_i = 0$ mod2), exponential speedup ($2^{n/2} \to O(n)$) to determine a





Shor '94 (p.70), $f(x) = f(x + r)$, resulting y is measured with probability $p(y) = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} \mathrm{e}^{2\pi i k r y/2^n}\right|^2$, gives $|y - 2^n/r| < 1/2$ with $p > .4$, sufficient to determine period r via partial fraction expansion, exponential speedup ($n2^n, \exp(n^{1/3}) \to O(n^3)$).
(Note: replaces $\mathbf{H}^{\otimes n}|x\rangle = \frac{1}{2^{n/2}}\sum_{0\le y<2^n} \mathrm{e}^{i\pi x\cdot y}|y\rangle$ with $\mathbf{U}_{\rm FT}|x\rangle = \frac{1}{2^{n/2}}\sum_{0\le y<2^n} \mathrm{e}^{2\pi i x y/2^n}|y\rangle$.)
Practical application is $f(x) \equiv b^x \bmod N$, where $b \equiv a^c \bmod N$ is an encrypted message, from which d', satisfying $cd' \equiv 1 \bmod r$, can be calculated, and d' recovers unencrypted message $a \equiv b^{d'} \bmod N$ (in contrast to using d, with $cd = 1 \bmod (p-1)(q-1)$, where $N = pq$ and r divides $(p-1)(q-1) = |G_{pq}|$).





Grover '96 (p.90), $f(x) = 1$ only for (m) marked value(s) $x = a$, uses "phase kickback" to express $\mathbf{U}_f$ in terms of $\mathbf{V} = \mathbf{1} - 2|a\rangle\langle a|$, and $\mathbf{W} = 2|\phi\rangle\langle\phi| - \mathbf{1} = \mathbf{H}^{\otimes n}(2|0\rangle\langle 0| - \mathbf{1})\mathbf{H}^{\otimes n}$ is easily constructed. Applying $\ell \approx \frac{\pi}{4}\frac{2^{n/2}}{\sqrt{m}}$ times gives probability $p(a) \approx 1 - O(m/2^n)$, for square-root speedup ($2^n/m \to \sqrt{2^n/m}$).

Simon's problem

exponential speedup

$f: n \to n-1$ bits

$f(x) = f(y)$ iff $x = y \oplus a$

[precursor to $f(x) = f(x+r)$]

Classically, how to determine a?

Try $x_0, x_1, x_2 \ldots$

if get lucky: $f(x_i) = f(x_j)$

$x_i = x_j \oplus a \Leftrightarrow a = x_j \oplus x_i$

but if not lucky, then we know $a \neq x_i \oplus x_j$ for any pair so far

So if we try m values $x_0 \ldots x_{m-1}$ then at most* we've excluded $\binom{m}{2} = \frac{m(m-1)}{2}$ values of a.

In order to exclude all but one value of a, how many values of $\{x_k\}$ necessary?

need $\frac{m(m-1)}{2} \approx 2^n$

$$\Rightarrow m \sim 2^{n/2}$$

*unless "carelessly" choose $x_\ell = x_i \oplus x_j \oplus x_k$ (previously chosen)
then $x_\ell \oplus x_{i,j,k}$ doesn't exclude any new pairs

classically to determine n bit a, need $\sim 2^{n/2}$ invocations of f.

Quantumly: need $O(n+\alpha)$

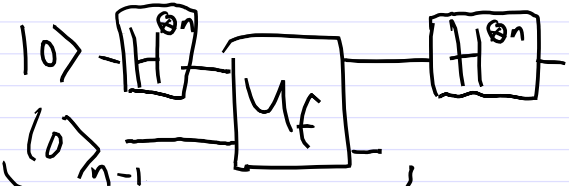
Eg. For $n=100$ $2^{n/2}=2^{50}\sim 10^{15}$

at 10M/sec $\rightarrow$ 3 yrs

with QM take only 120 invocations to get a (with probability $> 1-10^{-6}$)

$$U_f |x\rangle_n |y\rangle_{n-1} = |x\rangle_n |y \oplus f(x)\rangle_{n-1}$$

$$U_f H^{\otimes n} |x\rangle |0\rangle = \frac{1}{2^{n/2}} \sum_{0\le x<2^n} |x\rangle |f(x)\rangle$$



$f(x) = f(x \oplus a)$

$$\frac{1}{2^{n/2}} \sum_{0 \le x < 2^n} |x\rangle |f(x)\rangle$$

---

measure output, collapses to

$$\frac{1}{\sqrt{2}}\Big(|x_0\rangle + |x_0 \oplus a\rangle\Big)|f(x_0)\rangle$$

but only get one of two by measuring input (+ can't clone above state.)

But we can apply an operator <u>before</u> measuring inputs.

We renounce learning x_0, $x_0 \oplus a$ values but we can learn a <u>relation</u> between them: their mod 2 sum

$$H^{\otimes n} \frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_0 \oplus a\rangle\right)$$

$$= \frac{1}{2^{(n+1)/2}} \sum_y \left((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a)\cdot y}\right)|y\rangle$$

$$(-1)^{(x \oplus a) \cdot y} = (-1)^{x \cdot y}(-1)^{a \cdot y}$$

coeff of $|y\rangle$ is 0 if $a \cdot y = 1$

$(-1)^{x_0 \cdot y} - (-1)^{x_0 \cdot y} = 0$ else 2

So

$$= \frac{1}{2^{(n-1)/2}} \sum_{y \mid y \cdot a = 0} (-1)^{x_0 \cdot y} |y\rangle$$

(sum is over only y with $y \cdot a = 0$)

measure: gives some y s.t. $y \cdot a = 0$

Each such y constrains value of a to live in orthogonal subspace.

e.g. $n=3$, need to find $a=(a_2,a_1,a_0)$

measure $y=101$ $\quad a_0+a_2=0$

$y=010$ $\quad a_1=0$

since $a\neq 0$, constrains $a=101$

---

classically f once gives no info on a

f twice excludes one value of a.

quantumly f once excludes half the possible values, f again excludes half again, so twice excludes 3/4 of possible values

---

if really lucky can get first $n-1$ linearly independent (and non-zero) values of y, and hence determine a.

in general, need $n+\alpha$ values of y to have $> 1-\frac{1}{2^{\alpha+1}}$ probability of $n-1$ linearly independent