(Can be interpreted as "database search": want to find "marked element" x_0 in an unstructured database.)

Assume for now that $x_0 : f(x_0) = 1$ is <u>unique</u>.
(Generalization: later / homework)

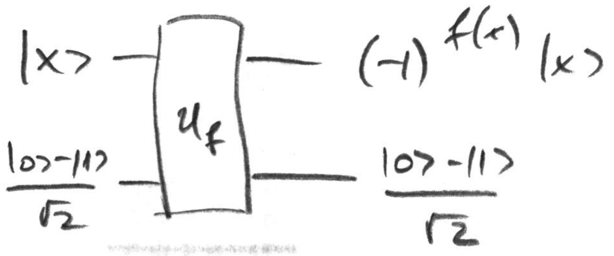
<u>Classically</u>: Need $O(N)$ queries to f for an <u>unstructured search</u> (i.e., w/out using properties of f).

<u>Quantum computer</u>: Will show that $O(\sqrt{N})$ queries enough.
(Note: Only quadratic speedup, but for a very large class of relevant problems)

<u>Ingredient 1</u>:

Oracle $O_f : |x\rangle \longmapsto (-1)^{f(x)} |x\rangle = (-1)^{\delta_{x,x_0}} |x\rangle$

Build via



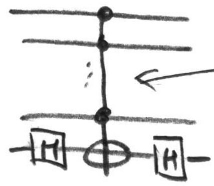
i.e., O_f flips amplitude of "marked" element.

Note that $O_f = I - 2 \cdot |x_0\rangle\langle x_0|$

Ingredient 2:

Unitary $O_0 : |x\rangle \longmapsto (-1)^{\delta_{x,0}} |x\rangle$

Corresponds to $C^{n-1}Z =$



n-qubit Toffoli

$\rightarrow$ can be realized efficiently

Again, $O_0 = I - 2|0\rangle\langle 0|$

Define $O_\omega := H^{\otimes n} O_0 H^{\otimes n} = I - 2|\omega\rangle\langle\omega|$; $|\omega\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$.

(Remark: We assumed here $N = 2^n$, but not necessary. Also, every search problem can be triv. embedded s.th. $N = 2^n$.)

Algorithm:

Start from $|\psi_0\rangle = |\omega\rangle = H^{\otimes n}|\underline{0}\rangle$.

Apply Grover iteration

$$G = -H^{\otimes n} O_0 H^{\otimes n} O_f = -O_\omega O_f :$$

$$|\psi_k\rangle \longmapsto |\psi_{k+1}\rangle = G|\psi_k\rangle = -O_\omega O_f |\psi_k\rangle.$$

<u>Observation</u>: Only 2 "special" vectors in O_f, O_ω:
$|x_0\rangle$ and $|\omega\rangle$ $\Rightarrow$ can analyze everything in two-dim. space spanned by $|x_0\rangle$ and $|\omega\rangle$!

Define $|\alpha\rangle := |x_0\rangle$

$$\left.\begin{array}{l} |\alpha\rangle := |x_0\rangle \\ |\beta\rangle := \frac{1}{\sqrt{N-1}} \sum_{x \neq x_0} |x\rangle \propto |\omega\rangle - \frac{1}{\sqrt{N}}|x_0\rangle \end{array}\right\} |\alpha\rangle \perp |\beta\rangle$$

We can always rewrite

$$a|\alpha\rangle + b|\beta\rangle = x|\omega\rangle + y|\omega^\perp\rangle, \quad \text{with } |\omega^\perp\rangle \perp |\omega\rangle$$

What is effect of O_f and $(-O_\omega)$?

$$O_f\left(a|\alpha\rangle + b|\beta\rangle\right) \underset{\uparrow}{=} -a|\alpha\rangle + b|\beta\rangle$$

$$O_f = I - 2|\alpha\rangle\langle\alpha|$$

$\Rightarrow$ <u>Reflection about $|\beta\rangle$!</u>

$$(-O_\omega)\left(x|\omega\rangle + y|\omega^\perp\rangle\right) = x|\omega\rangle - y|\omega^\perp\rangle$$

$\rightarrow$ <u>Reflection about $|\omega\rangle$!</u>

i.e.: Grover iteration = 1) reflect about $|\beta\rangle$

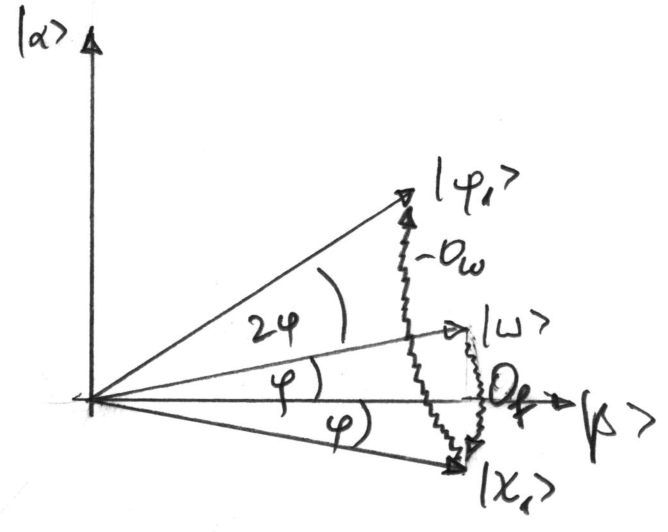
2) reflect about $|\omega\rangle$

So what happens in one iteration, if we start with $|\psi_0\rangle = |\omega\rangle$?

$$|\omega\rangle = \sin\varphi\,|\alpha\rangle + \cos\varphi\,|\beta\rangle$$

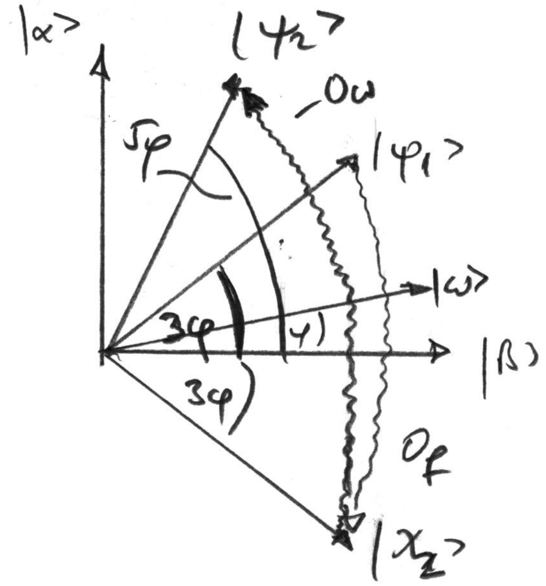
$$|\chi_1\rangle := O_f\,|\omega\rangle$$

$$|\psi_1\rangle = -O_\omega\,|\chi_1\rangle = -O_\omega\,O_f\,|\omega\rangle$$



$$|\psi_1\rangle = \sin(3\varphi)\,|\alpha\rangle + \cos(3\varphi)\,|\beta\rangle.$$

Next iteration: $$|\psi_2\rangle = -O_\omega\,\underbrace{O_f\,|\psi_1\rangle}_{=:\,|\chi_2\rangle}$$



$$\Rightarrow |\psi_2\rangle = \sin(5\varphi)\,|\alpha\rangle + \cos(5\varphi)\,|\beta\rangle$$

$$\Rightarrow |\psi_k\rangle = \sin\big((2k+1)\varphi\big)\,|\alpha\rangle + \cos\big((2k+1)\varphi\big)\,|\beta\rangle.$$

Want that $\varphi_k = (2k+1)\varphi \approx \frac{\pi}{2}$. Then, meas. will w/ high prob. yield $|\alpha\rangle = |k_0\rangle$!

<u>We have:</u> $|\omega\rangle = \frac{1}{\sqrt{N}}|\alpha\rangle + \sqrt{\frac{N-1}{N}}\,|\beta\rangle$

$$= \sin\varphi\,|\alpha\rangle + \cos\varphi\,|\beta\rangle$$

$$\Rightarrow \frac{\sin\varphi}{\cos\varphi} = \frac{\sqrt{\frac{1}{N}}}{\sqrt{\frac{N-1}{N}}} = \frac{1}{\sqrt{N-1}}$$

$$\Rightarrow \varphi \approx \frac{1}{\sqrt{N}} \text{ for large } N.$$

$\Rightarrow$ need $k \approx \frac{\pi}{4}\sqrt{N}$

$\Rightarrow$ $O(\sqrt{N})$ calls to f sufficient!

Quadratic speed-up w.r.t. classical algorithms

for general search problems!

Note: $\circ$ K solutions: Same method works with

$O\left(\sqrt{\frac{N}{K}}\right)$ steps ($\rightarrow$ HW)

- Can be adapted to case where K is unknown.

IV.4. The quantum Fourier transform, period finding, and Shor's factoring algorithm

Recall: Simon's algorithm $\rightarrow$ uses $H^{\otimes n} \hat{=}$ Fourier trafo over $\mathbb{Z}_2^{\otimes n}$ to find period in $\mathbb{Z}_2^{\otimes n}$.

$\rightarrow$ Can we construct a general Q. Fourier Trafo?

$\rightarrow$ Can it be implemented efficiently?

$\rightarrow$ Applications?

a) The Quantum Fourier Transform (QFT)

Fourier trafo (FT) on $\mathbb{C}^N$:

$$x = (x_0, \dots, x_{N-1}) \in \mathbb{C}^N$$
$$y = (y_0, \dots, y_{N-1}) \in \mathbb{C}^N$$

FT: $x \longmapsto y$ s.th. $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \cdot e^{2\pi i\, jk/N}$

Define QFT:

$$\boxed{|j\rangle \longmapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i\, jk/N} |k\rangle}$$

(Note: QFT: $\sum x_j |j\rangle \longmapsto \sum x_j e^{2\pi i jk/N} |k\rangle = \sum y_k |k\rangle$

$\Rightarrow$ QFT applies FT to amplitudes)

Computational cost of FT: $O(N^2)$ operations.

With $N = 2^n \rightarrow$ exp. cost in # of bits n.

Fast FT (FFT): $O(N \log N)$, but still $\sim \exp(n)$.

Will show: QFT can be implemented in $O(n^2)$ steps

$\longrightarrow$ exponential speed-up!

Rewrite QFT in binary:

* $N = 2^n$

* Write j in binary: $j = j_1 j_2 \cdots j_n = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0$

* Decimal point: $0.j_\ell j_{\ell+1} \cdots j_m = \frac{1}{2} j_\ell + \frac{1}{4} j_{\ell+1} + \dots$

Then:

$$|j\rangle \longmapsto \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i\, jk/2^n} \, |k\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{k_1=0,1} \cdots \sum_{k_n=0,1} e^{2\pi i\, j \left(\sum_{\ell=1}^{n} k_\ell 2^{-\ell} \right)} \, |k_1, \dots, k_n\rangle$$

$$= \bigotimes_{\ell=1}^{n} \left[\frac{1}{\sqrt{2}} \sum_{k_\ell=0,1} e^{2\pi i\, j k_\ell 2^{-\ell}} \, |k_\ell\rangle \right]$$

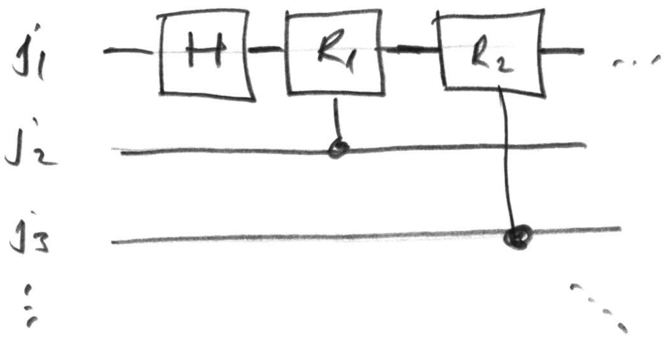
$$= \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \left[|0\rangle + e^{2\pi i\, \underbrace{j 2^{-\ell}}} \, |1\rangle \right]$$

$$\hookrightarrow j 2^{-\ell} = \underbrace{j_1 j_2 \cdots j_{n-\ell}}_{e^{2\pi i \cdot \text{integer}} = 1} \,.\, j_{n-\ell+1} \cdots j_n$$

$$= \frac{|0\rangle + e^{2\pi i\, 0.j_n} |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{2\pi i\, 0.j_{n-1} j_n} |1\rangle}{\sqrt{2}} \otimes \dots \otimes \frac{|0\rangle + e^{2\pi i\, 0.j_1 \cdots j_n} |1\rangle}{\sqrt{2}}$$

How to implement this map?

Start w/ right most term: $\frac{|0\rangle + e^{2\pi i\, 0.j_1 j_2 \dots j_n}|1\rangle}{\sqrt{2}}$



$$R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i \cdot 2^{-(d+1)}} \end{pmatrix}$$

action of circuit (up to norm.):

$H: |j_1\rangle \longmapsto |0\rangle + e^{2\pi i\, 0.j_1}|1\rangle$

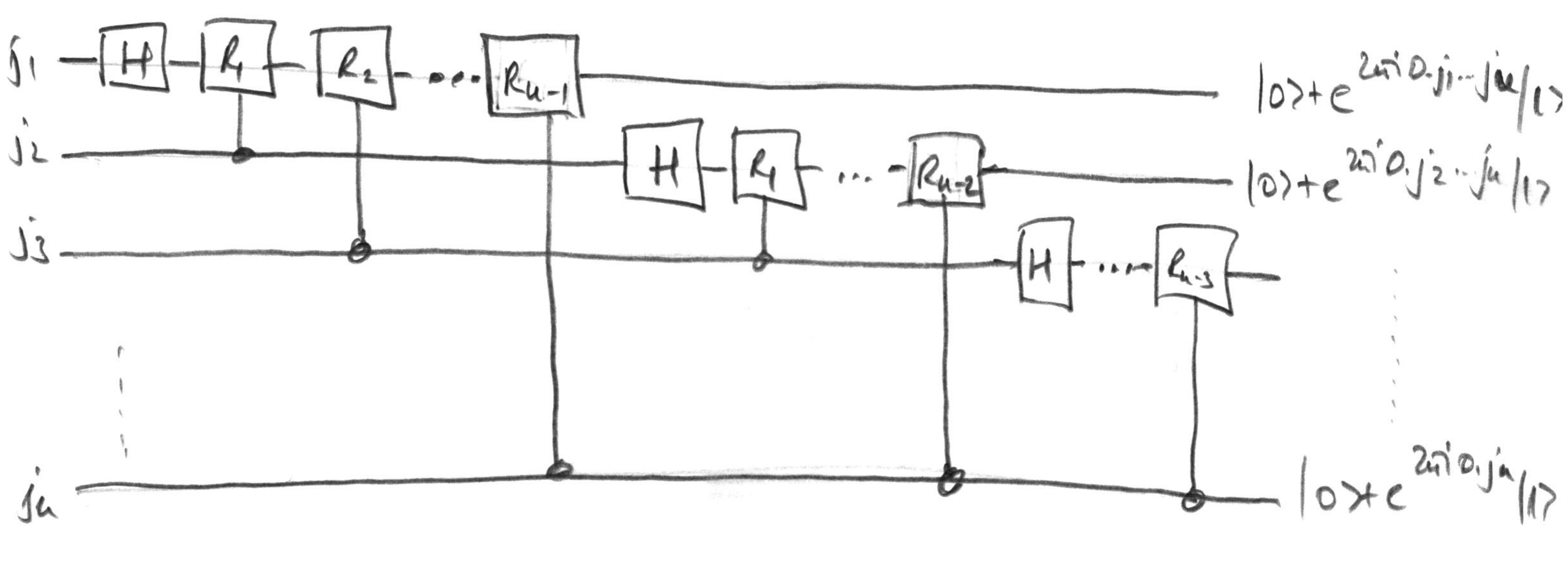
$C\text{-}R_1: \left(|0\rangle + e^{2\pi i\, 0.j_1}|1\rangle\right)|j_2\rangle \longmapsto \left(|0\rangle + e^{2\pi i\, 0.j_1 j_2}|1\rangle\right)|j_2\rangle$

$C\text{-}R_2: \left(|0\rangle + e^{2\pi i\, 0.j_1 j_2}|1\rangle\right)|j_2\rangle|j_3\rangle \longmapsto \left(|0\rangle + e^{2\pi i\, 0.j_1 j_2 j_3}|1\rangle\right)|j_2\rangle|j_3\rangle$

⋮ etc.

$\Longrightarrow$ obtain n-th qubit of QFT on 1st qubit.

Continue like that for $(j_2, \dots, j_n)$, $(j_3, \dots, j_n)$, ...



Gate count: $\frac{n(n+1)}{2} = O(n^2)$ gates!

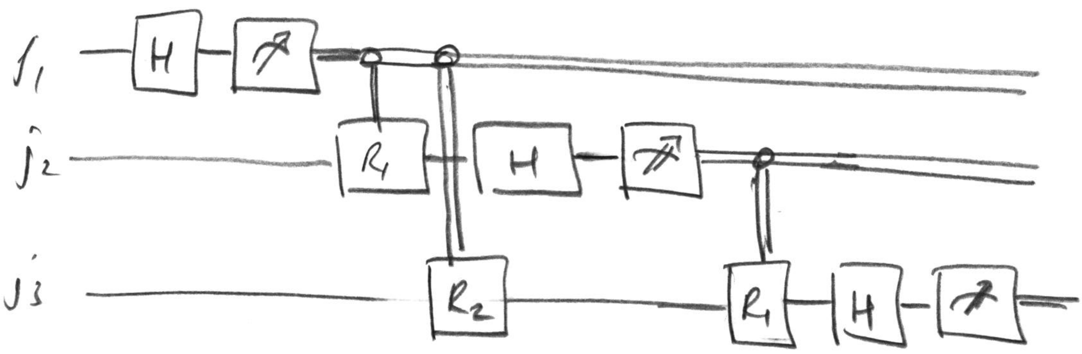
Notes: • Output is in reverse order (but reordering $\sim O(n)$ ops).

• 

$\Rightarrow$ can reverse C-R_d gates.

Upper (control) line acts as control in comp. basis:

If we measure after QFT, we can meas. after H & control R_d-gates classically!



$\Rightarrow$ only one-qubit gates needed!

6) Period finding

Use of QFT: period finding?

Consider $f : \{0,1\}^n \longrightarrow \{0,1\}^n$, s.th. $\exists r > 0$,

$f(x) = f(x+r)$ (and otherwise $f(x) \neq f(y)$)

Can we find r?

Use $\mathcal{U}_f : |x\rangle|y\rangle \longmapsto |x\rangle|y \oplus f(x)\rangle$

① $\frac{1}{2^{n/2}} \sum |x\rangle_A |0\rangle_B \xmapsto{\mathcal{U}_f} \frac{1}{2^{n/2}} \sum |x\rangle_A |f(x)\rangle_B$

② Measure $B \to A$ collapses to

$$\frac{1}{\sqrt{k_0}} \sum_{k=0}^{k_0} |x_0 + kr\rangle$$

(Note: As for Simon, we could omit this step!)

③ Apply QFT & measure in comp. basis:

$$\to \frac{1}{2^{n/2}\sqrt{k_0}} \sum_{k}^{k_0} \sum_{\ell=0}^{2^n-1} e^{2\pi i (x_0 + kr)\ell/2^n} |\ell\rangle$$

$$= \sum_{\ell=0}^{2^n-1} e^{2\pi i x_0 \ell/2^n} \underbrace{\left[\sum_{k=0}^{k_0-1} \frac{1}{2^{n/2}\sqrt{k_0}} e^{2\pi i k r \ell/2^n} \right]}_{=:\, a_\ell} |\ell\rangle$$

$|a_\ell|^2$: probability of outcome ℓ.

If $r \ll 2^n$: many values of k & almost periodic

$\Rightarrow$ $|a_\ell|^2$ peaked around ℓ s.th. $\frac{r\ell}{2^n} \approx$ integer

Explicit analysis of a_ℓ shows: w.h.p., we obtain ℓ s.th.
↳ "with high probability"

$$\frac{\ell}{2^n} \approx \frac{s}{r}$$

If $r \ll 2^n$, this can be used to determine $\frac{s}{r}$ w.h.p.

If s, r are coprime (i.e., $\gcd(s,r) = 1$) – this happens with large enough prob., related to density of primes – we can infer r!

$\Rightarrow$ Quantum Algorithm for period finding!

c) Application: factoring

One use of of period finding: factoring.

Given N (not prime) $\rightarrow$ find non-trivial r: $r | N$
↑ "divides"