$\Rightarrow O(\sqrt{N})$ steps sufficient!

$\Rightarrow$ Quadratic speed-up for general search problems!

Note: $\circ K$ solutions. Same method works with $O\left(\sqrt{\frac{N}{K}}\right)$ steps. ($\rightarrow$ HW).

$\circ$ Can also be adapted to the case when K is unknown.

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

Simon's alg.: Use $H^{\otimes n} \hat{=}$ Fourier trafo in $(\mathbb{Z}_2)^n$ to find period in $(\mathbb{Z}_2)^n$.

$\rightarrow$ Can we find general quantum F.T.?

$\rightarrow$ Can it find periods?

$\rightarrow$ Applications?

a) The Quantum Fourier Transform (QFT)

Fourier trafo (FT): $x = (x_0, \dots, x_{N-1}) \in \mathbb{C}^N$

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

y is FT of x: $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \cdot e^{2\pi i jk/N}$

QFT:

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

(Equiv.: $\sum_{j=0}^{N-1} x_j |j\rangle \longmapsto \sum_{j,k} x_j e^{2\pi i jk/N} |k\rangle = \sum_{k=0}^{N-1} y_k |k\rangle$)

Classical FT: $O(N^2)$ operations.

W.l.o.g. $N = 2^n \Rightarrow$ exp. scaling in # of bits n.

better: FFT (fast FT): $O(N \log N)$, but still exp. in n!

Will see: QFT can be implemented efficiently – in $O(n^2)$ steps!

Rewrite QFT:

- Use $N = 2^n$
- Write j in binary: $j = j_1 j_2 \cdots j_N = j_1 \cdot 2^{n-1} + j_2 \cdot 2^{n-2} + \cdots + j_n 2^0$
- Decimal point: $0.j_\ell j_{\ell+1} \cdots j_n = \frac{1}{2} j_\ell + \frac{1}{4} j_{\ell+1} + \cdots + \frac{1}{2^{n-\ell+1}} j_n$.

Then

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

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

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

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

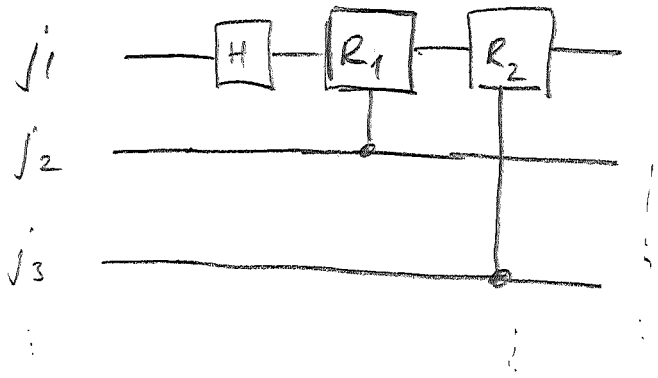
$$= \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \left[|0\rangle + \underbrace{e^{2\pi i j 2^{-\ell}}}_{\downarrow} |1\rangle \right]$$

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

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

How to build circuit?

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



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

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

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

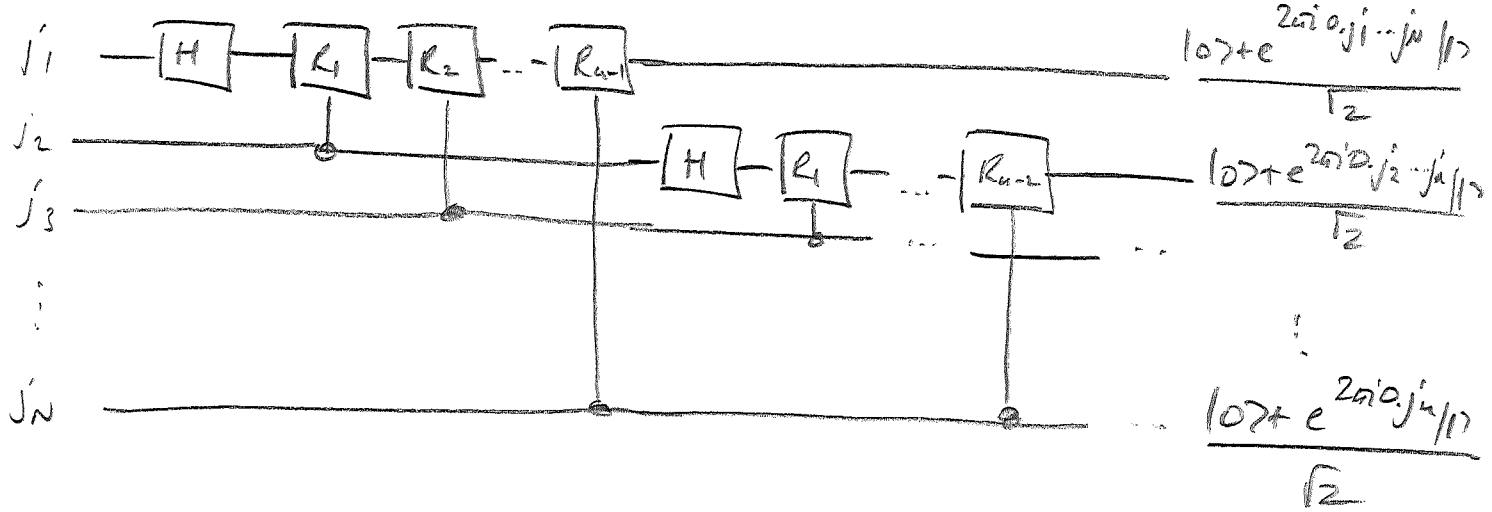
$$C\text{-}R_2 : \left(|0\rangle + e^{2\pi i\, 0.j_1 j_2}|1\rangle\right)|j_2\rangle|j_3\rangle \longmapsto \left(|0\rangle + e^{2\pi i\, 0.j_1 j_2 j_3}|1\rangle\right)|j_2\rangle|j_3\rangle$$

$\vdots$

etc.

$\Longrightarrow$ obtain with Result of QFT as 1st qubit!

Continue the steps:



# of gates: $\frac{n(n+1)}{2} = O(n^2) \Rightarrow$ efficient!

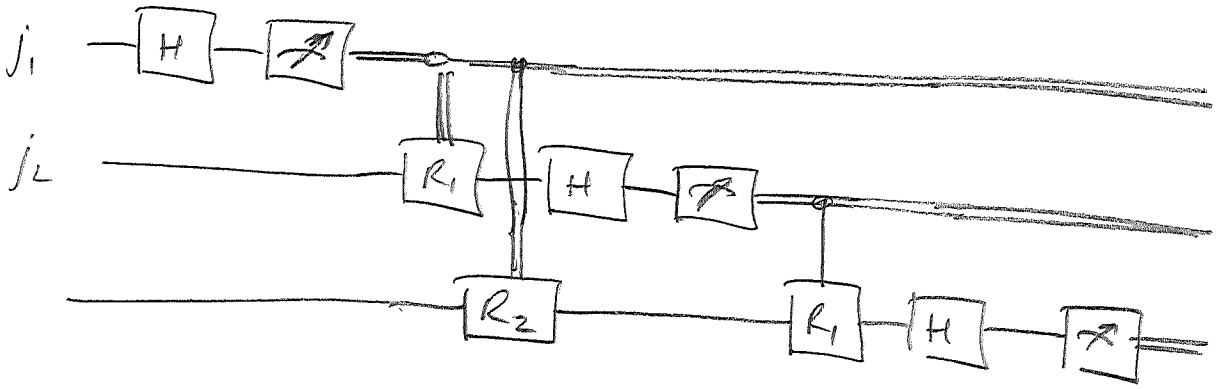
Notes: • Output qubits are in reverse order.

(Reordering: $O(n^2)$ operations.)

• Since  $\Rightarrow$ Rd only dep.

on classical value of upper qubit $\Rightarrow$

If we need to measure register a/fter QFT, we can measure a/fter H & control Rd from meas. outcome:



$\Rightarrow$ can be implemented w/ one-qubit gates only!

6) Period finding:

Use of QFT: finding periods (cf. Simon)

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

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

Can we find r? — Assume $r \ll 2^n$. ← "sufficiently small"

Use $U_f: |x\rangle_A |y\rangle_B \mapsto |x\rangle_A |y \oplus f(x)\rangle_B$

(1) $\frac{1}{2^{n/2}} \sum |x\rangle_A |0\rangle_B \xrightarrow{U_f} \frac{1}{2^{n/2}} \sum |x\rangle_A |f(x)\rangle_B$

(2) Measure $|f(x)\rangle_B$ — A collapses to

$$\frac{1}{\sqrt{k_0}} \sum_{k=0}^{k_0-1} |x_0 + kr\rangle \quad \left(\frac{2^n}{r} - 1 < k_0 \leq \frac{2^n}{r}\right)$$

(Note: As in Simon, we can omit this step!)

(3) Apply QFT:

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

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

Prob. distr. $|a_\ell|^2$ of meas. $|\ell\rangle$: Should be centered around ℓ s.th. $\frac{r\ell}{2^u} \approx$ integer!

More precisely:

Consider only ℓ s.th.

$$\ell = \frac{2^u}{r} \cdot s + \delta_s \; ; \; |\delta_s| \leq \frac{1}{2} ; \quad s = 0, \ldots, r-1$$

$$\Rightarrow a_\ell = \frac{1}{2^{u/2}\sqrt{k_0}} \sum_{k=0}^{k_0-1} e^{2\pi i k \left(\cancel{s} + \frac{r}{2^u}\delta_s \right)}$$

$$= \frac{1}{2^{u/2}\sqrt{k_0}} \; \frac{e^{2\pi i \cdot \frac{r}{2^u}\delta_s \cdot k_0} - 1}{e^{2\pi i \, r/2^u \delta_s} - 1}$$

$$\frac{2^u}{r} - 1 \leq k_0 \leq \frac{2^u}{r} \; ; \; r \ll 2^u \Rightarrow \frac{k_0 r}{2^u} \approx 1$$

$$\approx \frac{1}{2^{u/2}\sqrt{k_0}} \; \frac{e^{2\pi i \delta_s} - 1}{e^{2\pi i \, r/2^u \delta_s} - 1}$$

$\sin x \geq \frac{x}{\pi/2}$

$$\Rightarrow |a_\ell|^2 = \frac{1}{2^u k_0} \left(\frac{\sin(\pi \delta_s)}{\sin\left(\frac{\pi r}{2^u}\delta_s\right)} \right)^2 \geq \frac{1}{2^u k_0} \; \frac{\cancel{\pi^2 \delta_s^2}}{\cancel{\pi^2/4}\,\delta} \; \frac{\cancel{\pi^2 r^2}}{2^{2u}} \cancel{\delta_s^2}$$

($\sin\left(\frac{\pi r}{2^u}\delta_s\right) \ll 1$)

$$= \frac{4}{\pi^2 r} \cdot \frac{1}{\frac{k_0 r}{2^u}} = \frac{4}{\pi^2} \frac{1}{r}.$$

$\frac{k_0 r}{2^u} \approx 1$

Since $s = 0, \ldots, r-1$:

$$\text{Prob}\left(\left| \ell - \frac{2^u}{r} s \right| \leq \frac{1}{2} \right) \geq \frac{4}{\pi^2} \approx 0.41$$

$\Rightarrow$ with suff. high prob., we obtain an ℓ s.t.

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

$\Rightarrow$ can be used to determine $\frac{s}{r}$ w.h.p.

If s and r are coprime (i.e. $\gcd(r,s) = 1$) — happens w/ suff. high prob (can be shown e.g. using density of primes) — we can infer r!

$\Rightarrow$ Quantum algorithm for period finding!

c) Factoring:

One use of period finding: factoring.

Problem: Given N (not prime). Find non-trv. r : $r \mid N$.
(↑ divides.)

Algorithm:

(1) Select random a, $2 \leq a < N$.

If $\gcd(a,N) > 1 \Rightarrow$ done.

↑ cf. computable (Euclid's algorithm)

Assume $\gcd(a,N) = 1$.

(2) The smallest r with $a^r \bmod N = 1$ is called order of $a \bmod N$.

Note: Existence (of r) follows since $G = \{a \mid a < N, \gcd(a,N) = 1\}$ is a group: If $ab \bmod N = ab' \bmod N \Rightarrow$
$\Rightarrow a(b-b') \bmod N = 0 \Rightarrow N \mid b-b' \Rightarrow b = b'$.
Thus, $b \mapsto ab \bmod N$ is bijective, and thus $\exists b$ s.t. $ab \bmod N = 1$.

r is the period of $f_{N,a}(x) = a^x \bmod N$.

$f_{N,a}(x)$ can be computed efficiently:

With $x = x_{m-1} 2^{m-1} + x_{m-2} 2^{m-2} + \dots + x_0$,

$$a^x \bmod N = \underbrace{\left(a^{(2^{m-1})}\right)^{x_{m-1}}}_{\uparrow} \cdot \left(a^{(2^{m-2})}\right)^{x_{m-2}} \cdots a^{x_0} \bmod N,$$

eff. computable by $a \overset{\cdot^2}{\mapsto} a^2 \overset{\cdot^2}{\mapsto} (a^2)^2 \mapsto \dots$

$\Rightarrow$ r can be found eff. w/ a quantum computer!

(3) Assume r even:

$$a^r \bmod N = 1 \iff N \mid (a^r - 1) \iff$$

$$\iff N \mid (a^{r/2} - 1)(a^{r/2} + 1)$$

and $N \nmid (a^{r/2} - 1)$ (otherwise, $a^{r/2} \bmod N = 1$ ↯).

$\Rightarrow$ either $N \mid a^{r/2} + 1$

or N has non-triv. common factors with both $a^{r/2} \pm 1$

$\Rightarrow 1 \neq \gcd(N, a^{r/2} \pm 1) \mid N$

$\Rightarrow$ found non-trivial factor of N!

$\Rightarrow$ algorithm successful as long as

(i) r even and (ii) $N \nmid (a^{r/2}+1)$

$\Rightarrow$ can be shown to happen with $p \geq 1/k$ for random choice of a.

(Unless $N = p^k$, p prime $\rightarrow$ can be checked taking logs.)

$\Rightarrow$ efficient quantum algorithm for factoring!