$\Rightarrow$ 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: o 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 $\rightarrow$ use $H^{\otimes n} \hat{=}$ 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 $N = \exp(n)$.

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: $j = j_1 j_2 \dots j_n = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0$
* Decimal part: $0.j_\ell j_{\ell+1} \dots j_n = \frac{1}{2} j_\ell + \frac{1}{4} j_{\ell+1} + \dots$

Then:

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

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

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

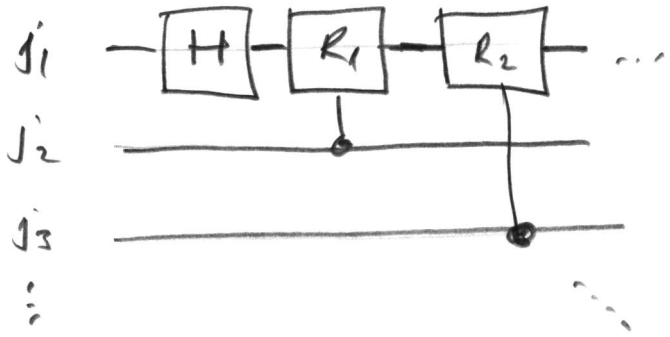
$$= \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \left[|0\rangle + e^{2\pi i j 2^{-\ell}} |1\rangle \right]$$

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

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

How to implement this map?

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



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

action of circuit (up to norm.):

$H_1: |j_1\rangle \longmapsto |0\rangle + e^{2\pi i\, 0.j_1}|1\rangle$

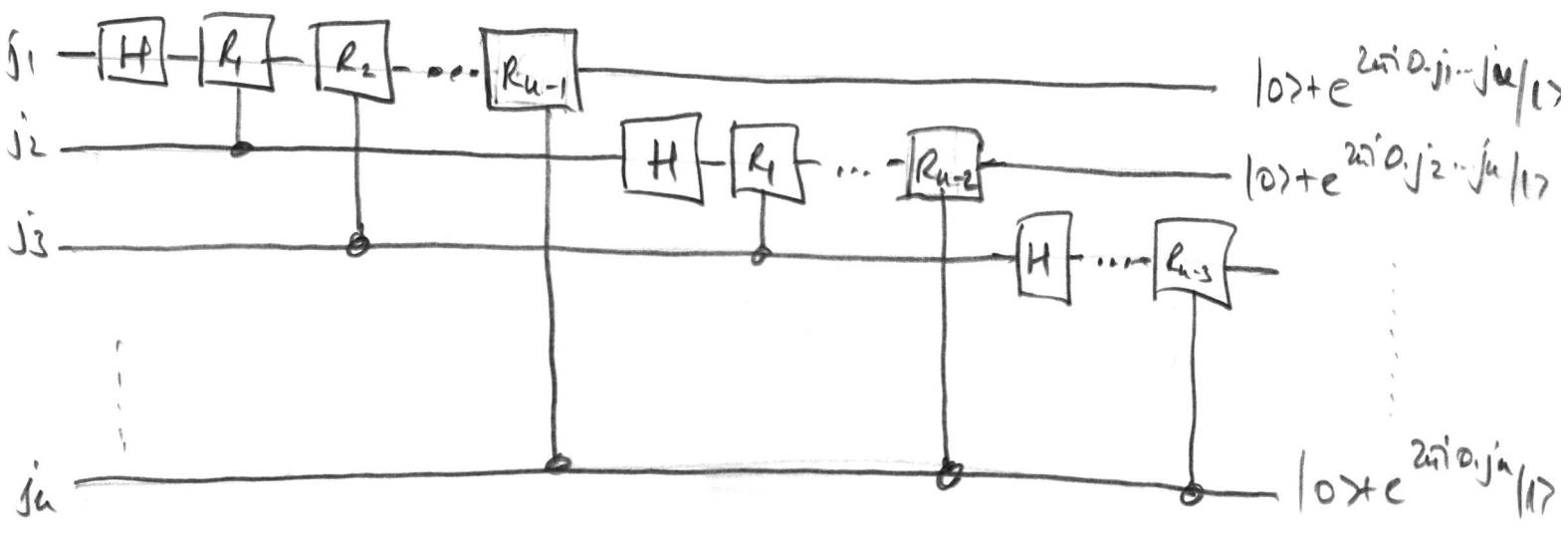
$c\text{-}R_1: (|0\rangle + e^{2\pi i\, 0.j_1}|1\rangle)|j_2\rangle \longmapsto (|0\rangle + e^{2\pi i\, 0.j_1 j_2}|1\rangle)|j_2\rangle$

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

$\vdots$ etc.

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

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



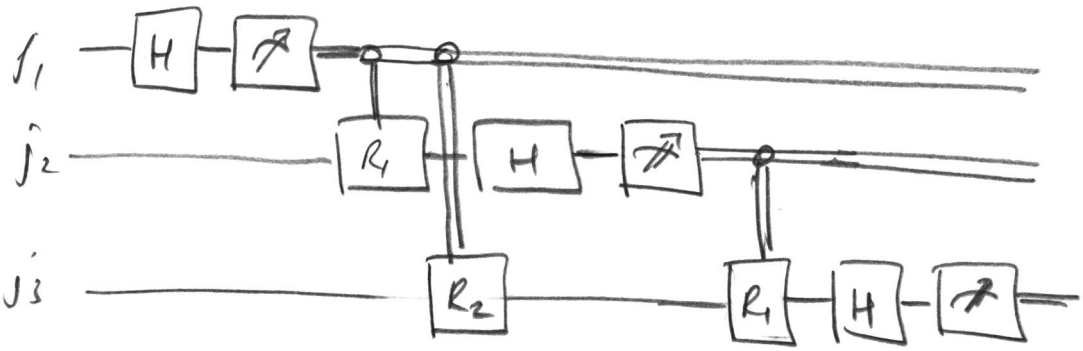
Gate count: $\frac{n(n+1)}{2} = O(n^2)$ gates !

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

•  ⇒ 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!



⇒ 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 $U_f : |x\rangle|y\rangle \longmapsto |x\rangle|y \oplus f(x)\rangle$

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

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: $r \mid N$

↑

"divides"

Algorithm:

① Select random a, $2 \leq a < N$.

If $\gcd(a,N) \geq 1 \Longrightarrow$ done!

↳ eff. computable! (Euclid's algorithm)

So assume $\gcd(a,N) = 1$.

② Let r be the smallest r s.th. $a^r \bmod N = 1$.

(Existence: (!) $\exists x,y: a^x \equiv a^y \bmod N \Rightarrow a^x(1-a^{y-x}) \equiv 0 \bmod N$

$\Rightarrow N \mid a^x(1-a^{y-x}) \overset{\gcd(a,N)=1}{\Longrightarrow} N \mid 1-a^{y-x} \Rightarrow a^{y-x} \equiv 1 \bmod N$)

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

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

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

$$a^x \bmod N = \underbrace{\left(a^{(2^{m-1})}\right)^{x_{m-1}} \cdot \left(a^{(2^{m-2})}\right)^{x_{m-2}} \cdot \dots}_{\text{eff. computable by } a \mapsto a^2 \mapsto a^4 \mapsto a^8 \mapsto \dots} \bmod N,$$

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

(3) Assume r even:

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

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-triv. factor of N!

$\Rightarrow$ Algorithm successful as long as
(i) r even & (ii) $N \nmid (a^{r/2} + 1)$

Can be shown to happen w/ $p \geq 1/2$ for random choice of a (unless $N = p^k$, p prime → can be checked by taking logs)

$\Rightarrow$ <u>efficient quantum algorithm for factoring</u>

<u>"Shor's algorithm"</u>

V. Quantum Error Correction

V.1. Introduction

- Coupling to environment reduces errors
- Classical computers: "macroscopic" → errors unlikely
- Q. computers: – need qubits = "single" quantum systems
  → fragile!
  – need coupling to "env." to realize gates!

So... can we protect quantum information from noise?

Classical error correction:

copy information, e.g. encode 1 bit in 3:

$$0 \longmapsto \hat{0} = 000$$
$$1 \longmapsto \hat{1} = 111$$

Bit flip w/ some (small) probability $p \Rightarrow$ typ. 0 or 1 bits flipped

Correction: majority vote, i.e.

$$000, 001, 010, 100 \longmapsto 000$$
$$111, 110, 101, 011 \longmapsto 111$$

$p_{error} = \text{prob}(\geq 2 \text{ flips}) = p^3 + 3p^2(1-p) \leq 3p^2 < p$ for $p < 1/3$!

⇒ effective error prob. decreased.

Can be improved by
- → using more bits
- → using smarter codes (encoding k bits)
- → nesting ("concatenating") codes

Quantum error correction:

Several potential problems:

- cannot copy qubits (and even if: how do we compare them?)
- different types of errors, e.g. X (bit flip) or Z (phase flip)
- errors can be continuous
- measuring qubits destroys q. info!

a) The 3-qubit bit flip code

Copy qubits in fixed steps:

$$|0\rangle \longmapsto |\hat{0}\rangle = |000\rangle$$

$$|1\rangle \longmapsto |\hat{1}\rangle = |111\rangle$$