Factoring integers with a quantum computer

Andrew Childs

Department of Combinatorics and Optimization
and Institute for Quantum Computing
University of Waterloo

Eighth Canadian Summer School on Quantum Information
Montreal, Quebec, 10 June 2008

The fundamental theorem of arithmetic

Every positive integer larger than 1 can be uniquely* factored as a product of prime numbers.

$$N = 2^{n_2} \times 3^{n_3} \times 5^{n_5} \times \cdots$$

* Up to the order of the factors.

Examples

$$15 =$$

Examples

$$15 = 3 \times 5$$

Examples

$$15 = 3 \times 5$$

$$91 =$$

Examples

$$15 = 3 \times 5$$

$$91 = 7 \times 13$$

Examples

$$15 = 3 \times 5$$

$$91 = 7 \times 13$$

$$\begin{array}{l}
3107418240490043721350750 \\
0358885679300373460228427 \\
2754572016194882320644051 \\
8081504556346829671723286 \\
7824379162728380334154710 \\
7310850191954852900733772 \\
4822783525742386454014691 \\
7366024776523466 09
\end{array} =$$

Examples

$$15 = 3 \times 5$$

$$91 = 7 \times 13$$

$$\begin{array}{l}
3107418240490043721350750 \\
0358885679300373460228427 \\
2754572016194882320644051 \\
8081504556346829671723286 \\
7824379162728380334154710 \\
7310850191954852900733772 \\
4822783525742386454014691 \\
7366024776523466 09
\end{array}
=
\begin{array}{l}
1634733645809253848 4 \\
4313388386509085984 1 \\
7836700330923121811 1 \\
0852389333100104508 1 \\
5121211816751157 9 \\
\times \\
1900871281664822113 1 \\
2685157393541397547 1 \\
8967899685154936666 3 \\
8539088027103802104 4 \\
989571912614655 71
\end{array}$$

"The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. It has engaged the industry and wisdom of ancient and modern geometers to such an extent that it would be superfluous to discuss the problem at length... Further, the dignity of the science itself seems to require that every possible means be explored for the solution of a problem so elegant and so celebrated."

– Carl Friedrich Gauss, *Disquisitiones Arithmeticæ* (1801)

RSA public-key cryptosystem

M

message

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

M

M

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

$n = pq$

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

$$n = pq$$

$$e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$$

encryption key

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

$$n = pq$$

$$e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$$

encryption key

$$d := e^{-1} \bmod (p-1)(q-1)$$

decryption key

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

n | n | $n = pq$

e | e | $e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$

encryption key

$$d := e^{-1} \bmod (p-1)(q-1)$$

decryption key

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

n | n | $n = pq$

e | e | $e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$

encryption key

$d := e^{-1} \bmod (p-1)(q-1)$

decryption key

$C := M^e \bmod n$

cyphertext

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

n | n | $n = pq$

e | e | $e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$

encryption key

$d := e^{-1} \bmod (p-1)(q-1)$

decryption key

$C := M^e \bmod n$

cyphertext

C | C

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M
message

primes p, q

n | n | $n = pq$

e | e | $e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$
encryption key

$d := e^{-1} \bmod (p-1)(q-1)$
decryption key

$C := M^e \bmod n$
cyphertext

C | C^d

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M
message

primes p, q

n | n | $n = pq$

e | e | $e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$
encryption key

$d := e^{-1} \bmod (p-1)(q-1)$
decryption key

$C := M^e \bmod n$
cyphertext | C | $C^d = M^{ed} \bmod n$

Rivest, Shamir, Adleman 1977

RSA public-key cryptosystem

M

message

primes p, q

n

n

$n = pq$

e

e

$e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$

encryption key

$d := e^{-1} \bmod (p-1)(q-1)$

decryption key

$C := M^e \bmod n$

cyphertext

C

$C^d = M^{ed} \bmod n$
$= M$

Rivest, Shamir, Adleman 1977

Outline

- Integers
- The quantum Fourier transform
- Period finding over $\mathbb{Z}_N$
- Period finding over $\mathbb{Z}$
- Reduction of factoring to period finding
- Beyond factoring

$$\mathbb{Z}$$

All else is the work of man

$$\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$$

All else is the work of man

$$\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$$

$$\begin{aligned}
\mathbb{Z}_N = \{ \quad & \bar{0} := \{\ldots, -N, 0, N, 2N, \ldots\} \\
& \bar{1} := \{\ldots, -N+1, 1, N+1, 2N+1, \ldots\} \\
& \bar{2} := \{\ldots, -N+2, 2, N+2, 2N+2, \ldots\} \\
& \quad \vdots \\
& \overline{N-1} := \{\ldots, -N-1, -1, N-1, 2N-1, \ldots\}\}
\end{aligned}$$

Multiplication modulo N

$$\mathbb{Z}_N^\times = \{j \in \mathbb{Z}_N : \exists k,\, jk = 1\}$$

Multiplication modulo N

$$\mathbb{Z}_N^\times = \{j \in \mathbb{Z}_N : \exists k,\, jk = 1\}$$

$$\varphi(N) := |\mathbb{Z}_N^\times|$$

Multiplication modulo N

$$\mathbb{Z}_N^\times = \{j \in \mathbb{Z}_N : \exists k,\, jk = 1\}$$

$$\varphi(N) := |\mathbb{Z}_N^\times|$$

$$\varphi(p) = p - 1$$

Multiplication modulo N

$$\mathbb{Z}_N^\times = \{j \in \mathbb{Z}_N : \exists k,\, jk = 1\}$$

$$\varphi(N) := |\mathbb{Z}_N^\times|$$

$$\varphi(p) = p - 1$$

$$\varphi(p^n) = (p-1)p^{n-1}$$

Multiplication modulo N

$$\mathbb{Z}_N^\times = \{j \in \mathbb{Z}_N : \exists k,\, jk = 1\}$$

$$\varphi(N) := |\mathbb{Z}_N^\times|$$

$$\varphi(p) = p - 1$$

$$\varphi(p^n) = (p-1)p^{n-1}$$

$$\varphi(p_1^{n_1} \cdots p_k^{n_k}) = (p_1 - 1)p_1^{n_1 - 1} \cdots (p_k - 1)p_k^{n_k - 1}$$

Multiplication modulo N

$$\mathbb{Z}_N^\times = \{j \in \mathbb{Z}_N : \exists k,\, jk = 1\}$$

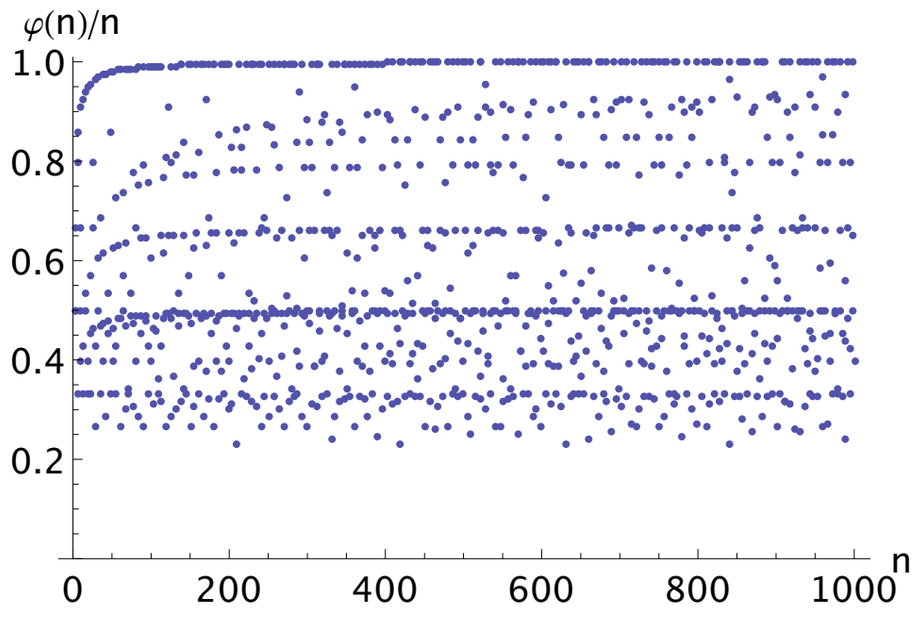
$$\varphi(N) := |\mathbb{Z}_N^\times|$$

$$\varphi(p) = p - 1$$

$$\varphi(p^n) = (p-1)p^{n-1}$$

$$\varphi(p_1^{n_1} \cdots p_k^{n_k}) = (p_1 - 1)p_1^{n_1 - 1} \cdots (p_k - 1)p_k^{n_k - 1}$$

Fact: $$\frac{\varphi(N)}{N} = \Omega\left(\frac{1}{\log\log N}\right)$$



The quantum Fourier transform

Quantum Fourier transform over $\mathbb{Z}_N$

$$\omega_N := e^{2\pi \mathrm{i}/N}$$

$$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

Quantum Fourier transform over $\mathbb{Z}_N$

$$\omega_N := e^{2\pi \mathrm{i}/N}$$

$$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$F_{\mathbb{Z}_N} = \frac{1}{\sqrt{N}} \sum_{x,y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle\langle x|$$

Quantum Fourier transform over $\mathbb{Z}_N$

$$\omega_N := e^{2\pi \mathrm{i}/N}$$

$$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$F_{\mathbb{Z}_N} = \frac{1}{\sqrt{N}} \sum_{x,y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle\langle x|$$

$$= \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2N-2} & \cdots & \omega_N^{(N-1)(N-1)} \end{pmatrix}$$

Examples

Examples

$$F_{\mathbb{Z}_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H \quad \text{(Hadamard gate)}$$

Examples

$$F_{\mathbb{Z}_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H \quad \text{(Hadamard gate)}$$

$$F_{\mathbb{Z}_3} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{pmatrix}$$

Examples

$$F_{\mathbb{Z}_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H \quad \text{(Hadamard gate)}$$

$$F_{\mathbb{Z}_3} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{pmatrix}$$

$$F_{\mathbb{Z}_6} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_6 & \omega_6^2 & \omega_6^3 & \omega_6^4 & \omega_6^5 \\ 1 & \omega_6^2 & \omega_6^4 & 1 & \omega_6^2 & \omega_6^4 \\ 1 & \omega_6^3 & 1 & \omega_6^3 & 1 & \omega_6^3 \\ 1 & \omega_6^4 & \omega_6^2 & 1 & \omega_6^4 & \omega_6^2 \\ 1 & \omega_6^5 & \omega_6^4 & \omega_6^3 & \omega_6^2 & \omega_6 \end{pmatrix}$$

Examples

$$F_{\mathbb{Z}_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H \quad \text{(Hadamard gate)}$$

$$F_{\mathbb{Z}_3} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{pmatrix}$$

$$F_{\mathbb{Z}_6} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_6 & \omega_6^2 & \omega_6^3 & \omega_6^4 & \omega_6^5 \\ 1 & \omega_6^2 & \omega_6^4 & 1 & \omega_6^2 & \omega_6^4 \\ 1 & \omega_6^3 & 1 & \omega_6^3 & 1 & \omega_6^3 \\ 1 & \omega_6^4 & \omega_6^2 & 1 & \omega_6^4 & \omega_6^2 \\ 1 & \omega_6^5 & \omega_6^4 & \omega_6^3 & \omega_6^2 & \omega_6 \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -\omega_3^2 & \omega_3 & -1 & \omega_3^2 & -\omega_3 \\ 1 & \omega_3 & \omega_3^2 & 1 & \omega_3 & \omega_3^2 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \omega_3^2 & \omega_3 & 1 & \omega_3^2 & \omega_3 \\ 1 & -\omega_3 & \omega_3^2 & -1 & \omega_3 & -\omega_3^2 \end{pmatrix}$$

Examples

$$F_{\mathbb{Z}_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H \quad \text{(Hadamard gate)}$$

$$F_{\mathbb{Z}_3} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{pmatrix}$$

$$F_{\mathbb{Z}_6} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_6 & \omega_6^2 & \omega_6^3 & \omega_6^4 & \omega_6^5 \\ 1 & \omega_6^2 & \omega_6^4 & 1 & \omega_6^2 & \omega_6^4 \\ 1 & \omega_6^3 & 1 & \omega_6^3 & 1 & \omega_6^3 \\ 1 & \omega_6^4 & \omega_6^2 & 1 & \omega_6^4 & \omega_6^2 \\ 1 & \omega_6^5 & \omega_6^4 & \omega_6^3 & \omega_6^2 & \omega_6 \end{pmatrix} = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -\omega_3^2 & \omega_3 & -1 & \omega_3^2 & -\omega_3 \\ 1 & \omega_3 & \omega_3^2 & 1 & \omega_3 & \omega_3^2 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \omega_3^2 & \omega_3 & 1 & \omega_3^2 & \omega_3 \\ 1 & -\omega_3 & \omega_3^2 & -1 & \omega_3 & -\omega_3^2 \end{pmatrix}$$

$$\cong F_{\mathbb{Z}_2} \otimes F_{\mathbb{Z}_3}$$

by the isomorphism $\mathbb{Z}_6 \cong \mathbb{Z}_2 \times \mathbb{Z}_3$

QFT implementation ($N = 2^n$)

Write $x = \sum_{j=0}^{n-1} 2^j x_j,\ y = \sum_{j=0}^{n-1} 2^j y_j.$

QFT implementation ($N = 2^n$)

Write $x = \sum_{j=0}^{n-1} 2^j x_j,\ y = \sum_{j=0}^{n-1} 2^j y_j.$

$$F_{\mathbb{Z}_{2^n}} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \omega_{2^n}^{x(\sum_{j=0}^{n-1} 2^j y_j)} |y_0\rangle \otimes \cdots \otimes |y_{n-1}\rangle$$

QFT implementation ($N = 2^n$)

Write $x = \sum_{j=0}^{n-1} 2^j x_j,\ y = \sum_{j=0}^{n-1} 2^j y_j.$

$$F_{\mathbb{Z}_{2^n}} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \omega_{2^n}^{x\left(\sum_{j=0}^{n-1} 2^j y_j\right)} |y_0\rangle \otimes \cdots \otimes |y_{n-1}\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \bigotimes_{j=0}^{n-1} \omega_{2^n}^{x y_j 2^j} |y_j\rangle$$

QFT implementation ($N = 2^n$)

Write $x = \sum_{j=0}^{n-1} 2^j x_j,\ y = \sum_{j=0}^{n-1} 2^j y_j.$

$$F_{\mathbb{Z}_{2^n}} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \omega_{2^n}^{x\left(\sum_{j=0}^{n-1} 2^j y_j\right)} |y_0\rangle \otimes \cdots \otimes |y_{n-1}\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \bigotimes_{j=0}^{n-1} \omega_{2^n}^{x y_j 2^j} |y_j\rangle$$

$$= \frac{1}{\sqrt{2^n}} \bigotimes_{j=0}^{n-1} \sum_{y_j \in \{0,1\}} e^{2\pi \mathrm{i}\, x y_j / 2^{n-j}} |y_j\rangle$$

QFT implementation ($N = 2^n$)

Write $x = \sum_{j=0}^{n-1} 2^j x_j,\ y = \sum_{j=0}^{n-1} 2^j y_j.$

$$F_{\mathbb{Z}_{2^n}} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \omega_{2^n}^{x\left(\sum_{j=0}^{n-1} 2^j y_j\right)} |y_0\rangle \otimes \cdots \otimes |y_{n-1}\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \bigotimes_{j=0}^{n-1} \omega_{2^n}^{x y_j 2^j} |y_j\rangle$$

$$= \frac{1}{\sqrt{2^n}} \bigotimes_{j=0}^{n-1} \sum_{y_j \in \{0,1\}} e^{2\pi \mathrm{i}\, x y_j / 2^{n-j}} |y_j\rangle$$

$$= \bigotimes_{j=0}^{n-1} \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi \mathrm{i} x 2^j / 2^n} |1\rangle)$$

QFT implementation ($N = 2^n$)

Write $x = \sum_{j=0}^{n-1} 2^j x_j,\ y = \sum_{j=0}^{n-1} 2^j y_j.$

$$F_{\mathbb{Z}_{2^n}} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \omega_{2^n}^{x\left(\sum_{j=0}^{n-1} 2^j y_j\right)} |y_0\rangle \otimes \cdots \otimes |y_{n-1}\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \bigotimes_{j=0}^{n-1} \omega_{2^n}^{x y_j 2^j} |y_j\rangle$$

$$= \frac{1}{\sqrt{2^n}} \bigotimes_{j=0}^{n-1} \sum_{y_j \in \{0,1\}} e^{2\pi \mathrm{i}\, x y_j / 2^{n-j}} |y_j\rangle$$

$$= \bigotimes_{j=0}^{n-1} \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi \mathrm{i} x 2^j / 2^n} |1\rangle)$$

$$= \bigotimes_{j=0}^{n-1} \frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi \mathrm{i} \sum_{k=0}^{n-1} 2^{j+k-n} x_k} |1\rangle)$$

QFT circuit

We have $F_{\mathbb{Z}_{2^n}} |x\rangle = \bigotimes_{j=0}^{n-1} |z_j\rangle$ where

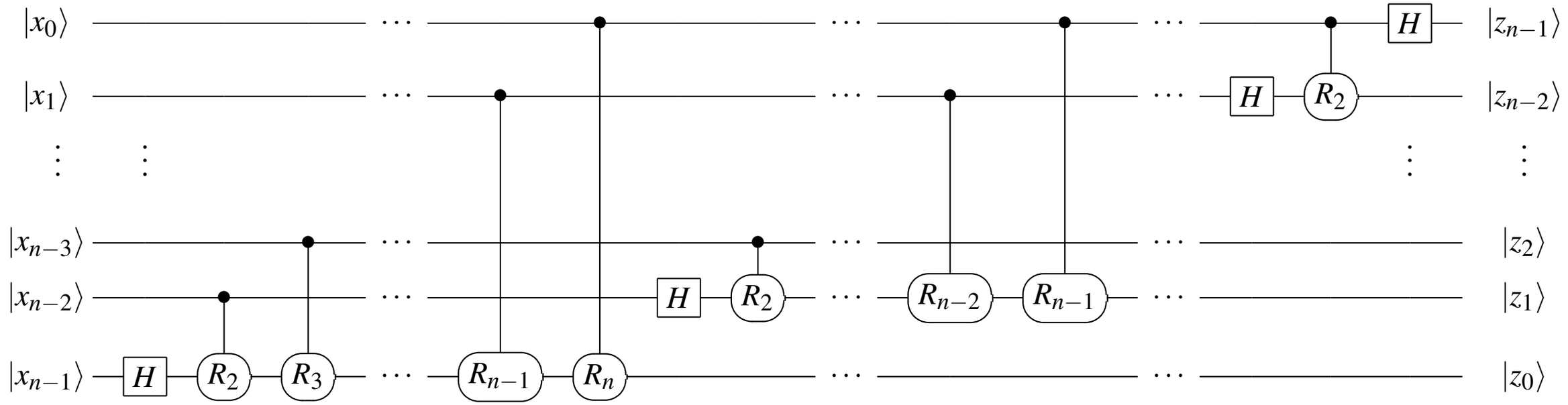
$$|z_j\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi \mathrm{i}(2^{j-n}x_0 + 2^{j+1-n}x_1 + \cdots + 2^{-1}x_{n-1-j})}|1\rangle)$$

QFT circuit

We have $F_{\mathbb{Z}_{2^n}} |x\rangle = \bigotimes_{j=0}^{n-1} |z_j\rangle$ where

$$|z_j\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi \mathrm{i}(2^{j-n}x_0 + 2^{j+1-n}x_1 + \cdots + 2^{-1}x_{n-1-j})}|1\rangle)$$

Quantum circuit:



$$R_r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi \mathrm{i}/2^r} \end{pmatrix}$$

note reversal of order

More general QFTs

There are efficient quantum circuits for $F_{\mathbb{Z}_N}$ for general N; see

- Kitaev, *Quantum measurements and the abelian stabilizer problem*, quant-ph/9511026
- Hales and Hallgren, *An improved quantum Fourier transform algorithm and applications*, FOCS 2000

More general QFTs

There are efficient quantum circuits for $F_{\mathbb{Z}_N}$ for general N; see

- Kitaev, *Quantum measurements and the abelian stabilizer problem*, quant-ph/9511026
- Hales and Hallgren, *An improved quantum Fourier transform algorithm and applications*, FOCS 2000

In general, $F_{\mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_k}} = F_{\mathbb{Z}_{N_1}} \otimes \cdots \otimes F_{\mathbb{Z}_{N_k}}$

More general QFTs

There are efficient quantum circuits for $F_{\mathbb{Z}_N}$ for general N; see

- Kitaev, *Quantum measurements and the abelian stabilizer problem*, quant-ph/9511026
- Hales and Hallgren, *An improved quantum Fourier transform algorithm and applications*, FOCS 2000

In general, $F_{\mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_k}} = F_{\mathbb{Z}_{N_1}} \otimes \cdots \otimes F_{\mathbb{Z}_{N_k}}$

Example: $F_{(\mathbb{Z}_2)^n} = H \otimes \cdots \otimes H$

More general QFTs

There are efficient quantum circuits for $F_{\mathbb{Z}_N}$ for general N; see

- Kitaev, *Quantum measurements and the abelian stabilizer problem*, quant-ph/9511026
- Hales and Hallgren, *An improved quantum Fourier transform algorithm and applications*, FOCS 2000

In general, $F_{\mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_k}} = F_{\mathbb{Z}_{N_1}} \otimes \cdots \otimes F_{\mathbb{Z}_{N_k}}$

Example: $F_{(\mathbb{Z}_2)^n} = H \otimes \cdots \otimes H$

We can also define a Fourier transform for a nonabelian group; many of these can be implemented efficiently as well.

Period finding over $\mathbb{Z}_N$

Periodic functions

A function $f : \mathbb{Z}_N \to S$ is *periodic with period* $r \in \mathbb{Z}_N$ if

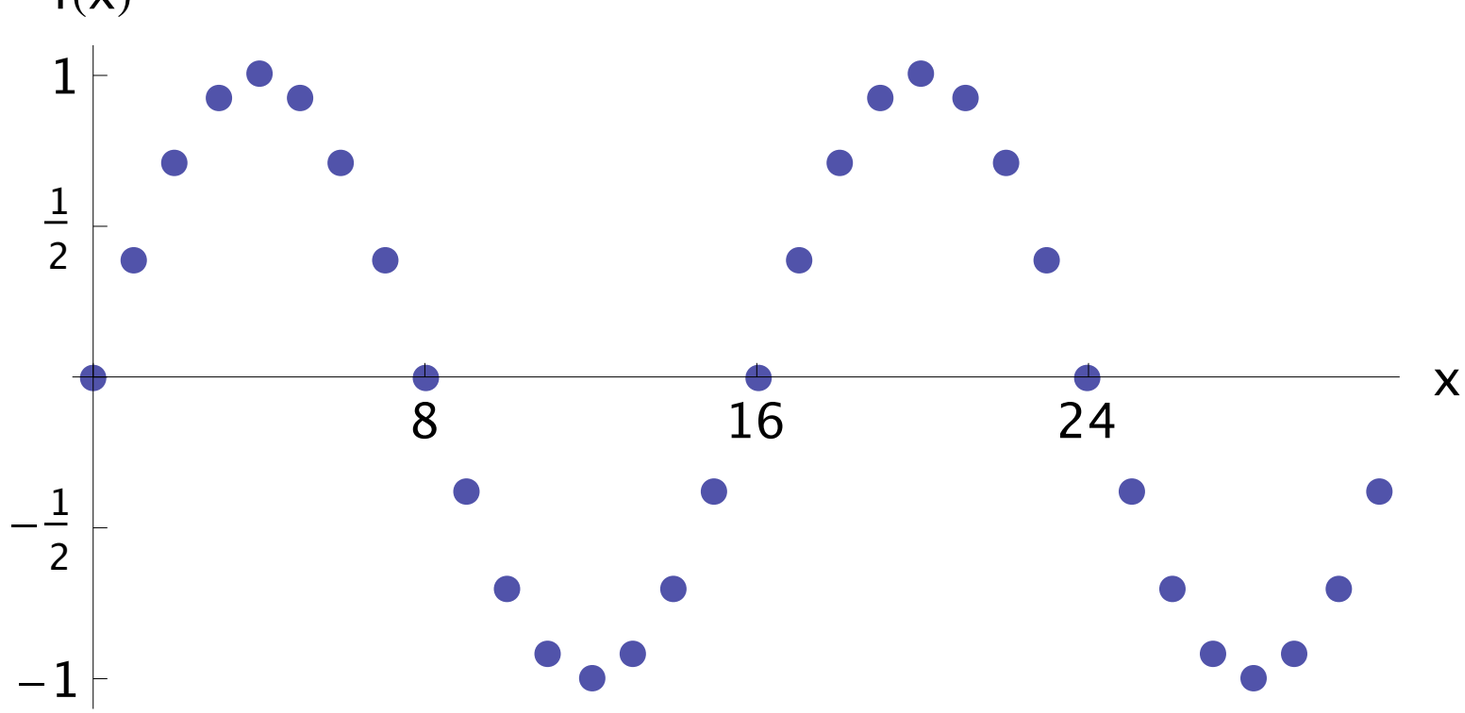
$$f(x) = f(y) \quad \text{if} \quad \frac{x-y}{r} \in \mathbb{Z}$$

Periodic functions

A function $f : \mathbb{Z}_N \to S$ is *periodic with period* $r \in \mathbb{Z}_N$ if

$$f(x) = f(y) \quad \text{if} \quad \frac{x-y}{r} \in \mathbb{Z}$$

Example (N=32):

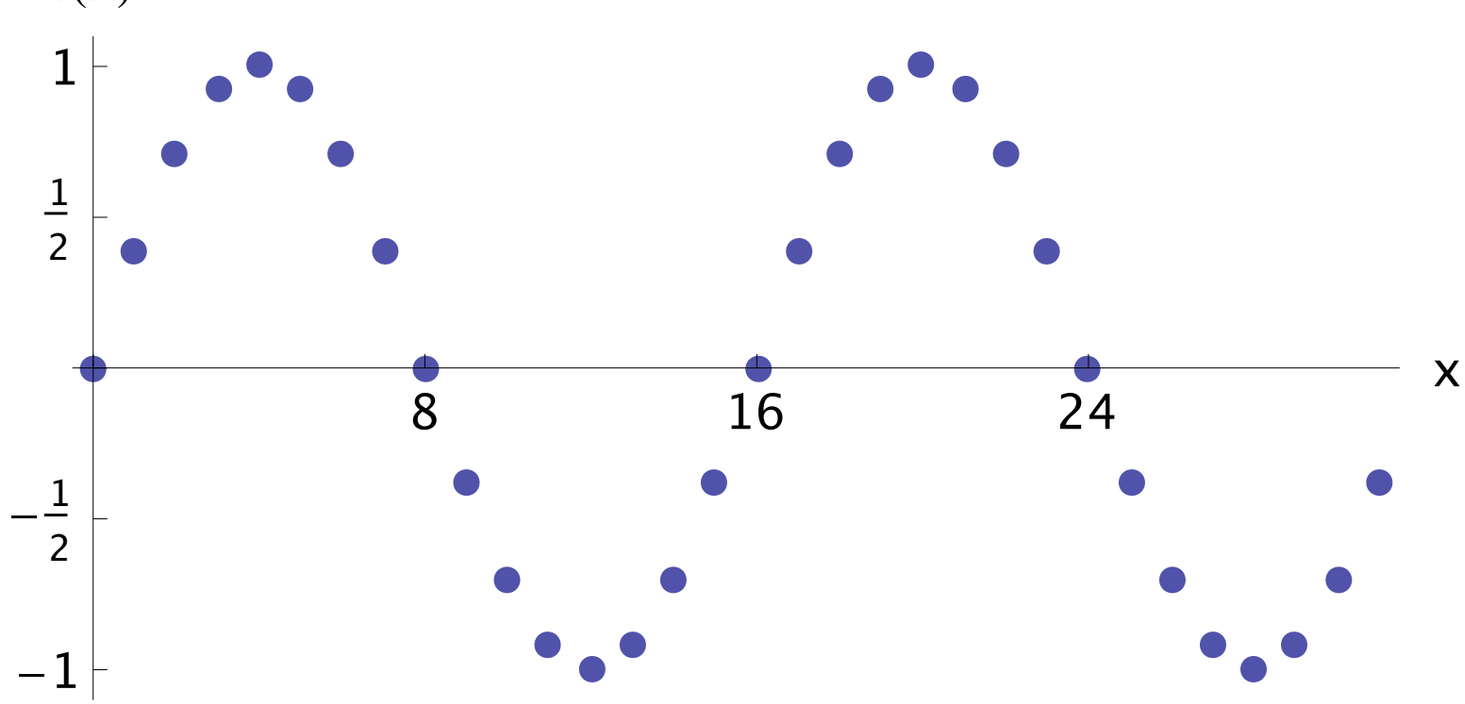


Periodic functions

A function $f : \mathbb{Z}_N \to S$ is *periodic with period* $r \in \mathbb{Z}_N$ if

$$f(x) = f(y) \quad \text{if} \quad \frac{x-y}{r} \in \mathbb{Z}$$

Example (N=32):



(Notice that r must divide N.)

Periodic functions

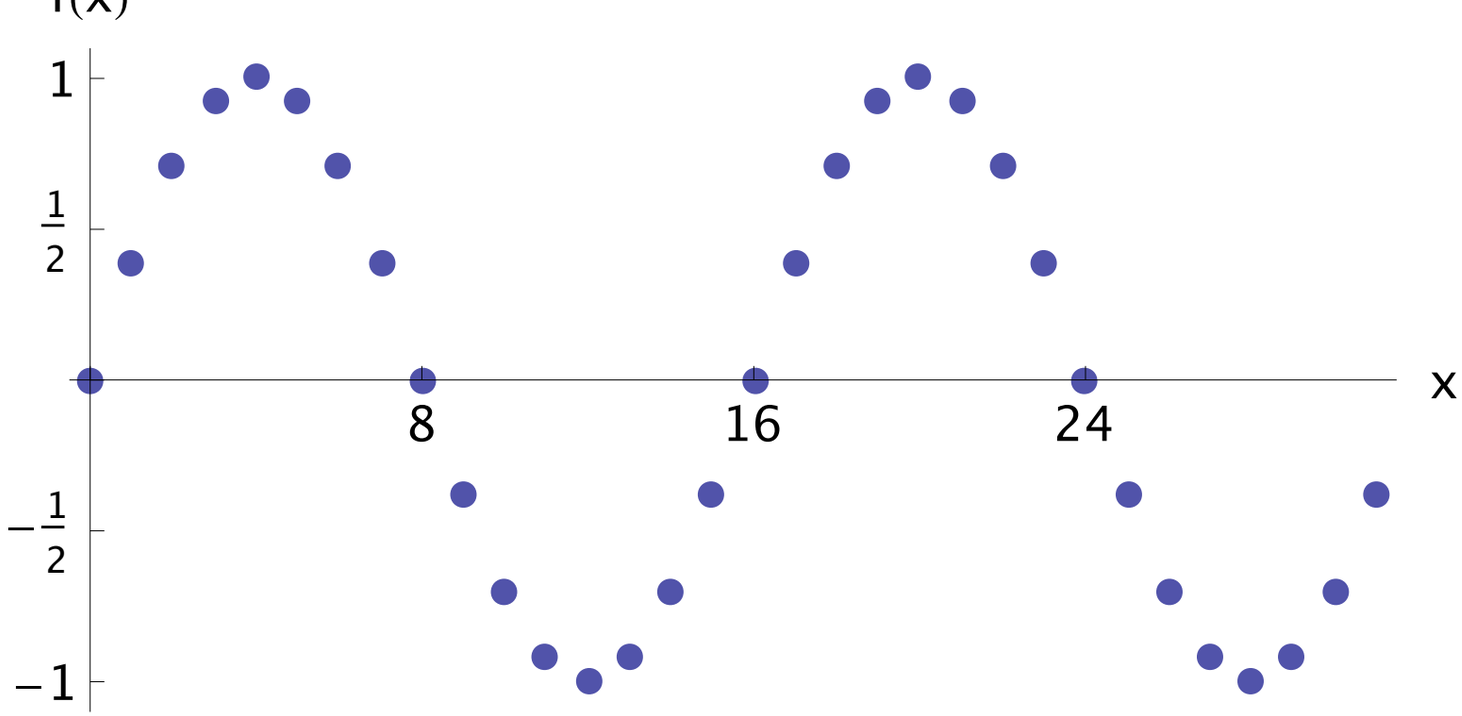
A function $f : \mathbb{Z}_N \to S$ is *injectively periodic with period* $r \in \mathbb{Z}_N$ if

$$f(x) = f(y) \text{ if and only if } \frac{x-y}{r} \in \mathbb{Z}$$

Periodic functions

A function $f : \mathbb{Z}_N \to S$ is *injectively periodic with period* $r \in \mathbb{Z}_N$ if

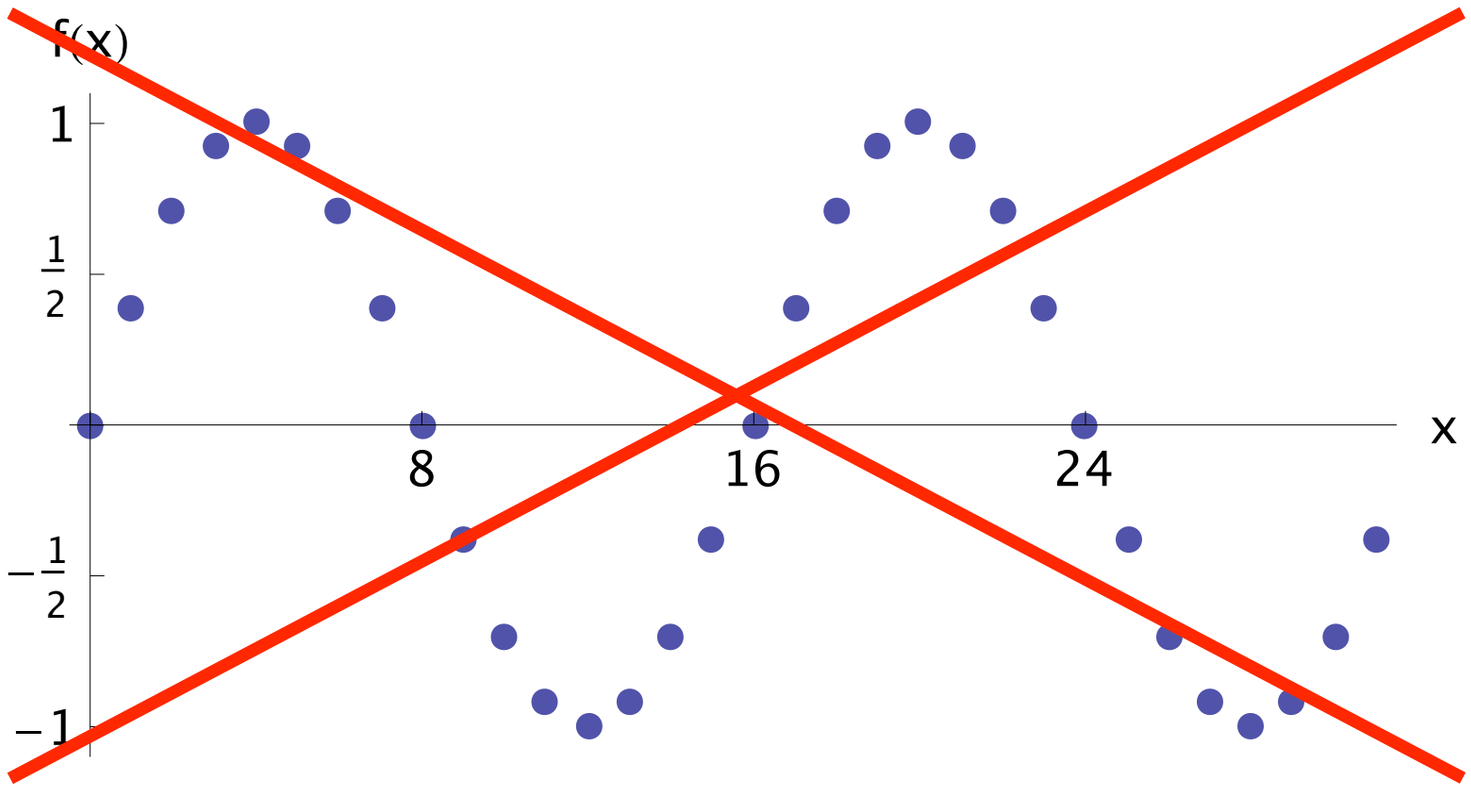
$$f(x) = f(y) \text{ if and only if } \frac{x-y}{r} \in \mathbb{Z}$$



Periodic functions

A function $f : \mathbb{Z}_N \to S$ is *injectively periodic with period* $r \in \mathbb{Z}_N$ if

$$f(x) = f(y) \text{ if and only if } \frac{x-y}{r} \in \mathbb{Z}$$



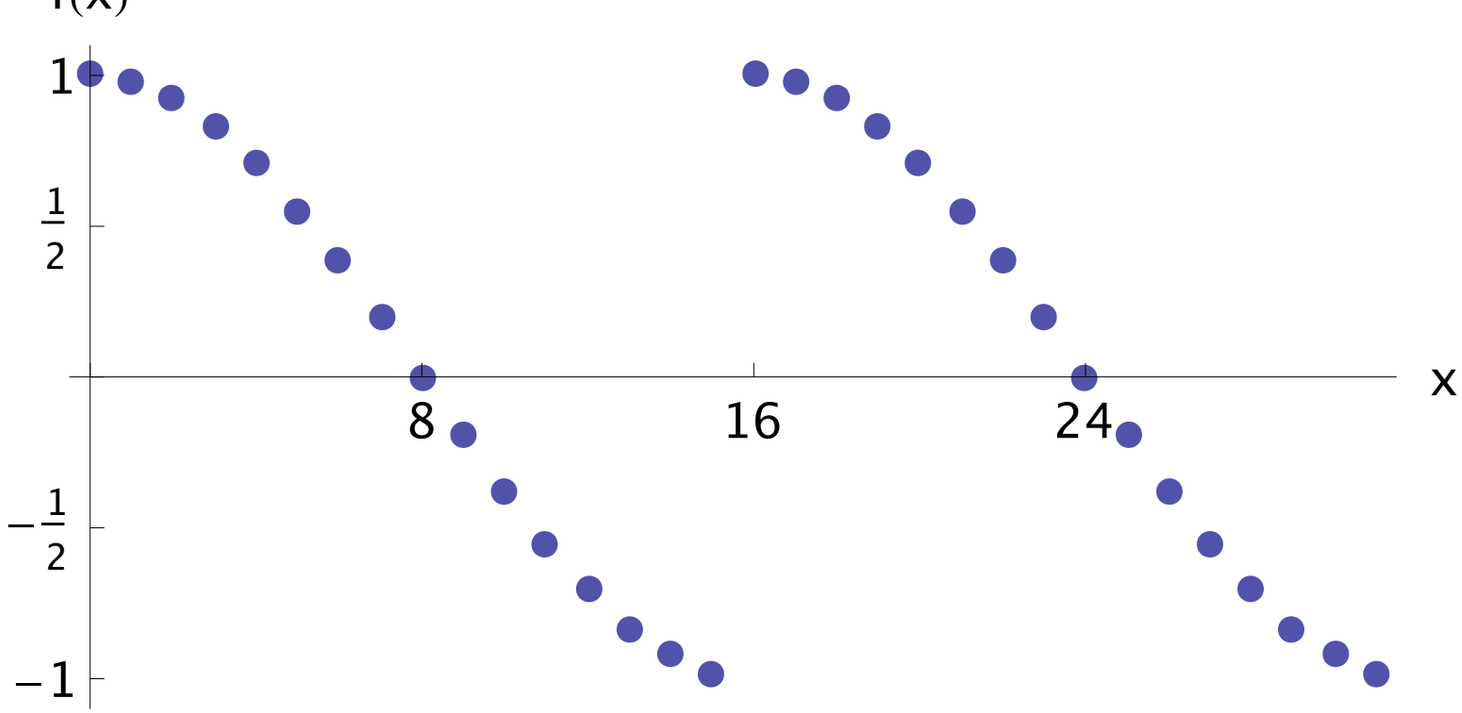
periodic, but not *injectively* periodic

Periodic functions

A function $f : \mathbb{Z}_N \to S$ is *injectively periodic with period* $r \in \mathbb{Z}_N$ if

$$f(x) = f(y) \text{ if and only if } \frac{x-y}{r} \in \mathbb{Z}$$

Example ($N = 32$):

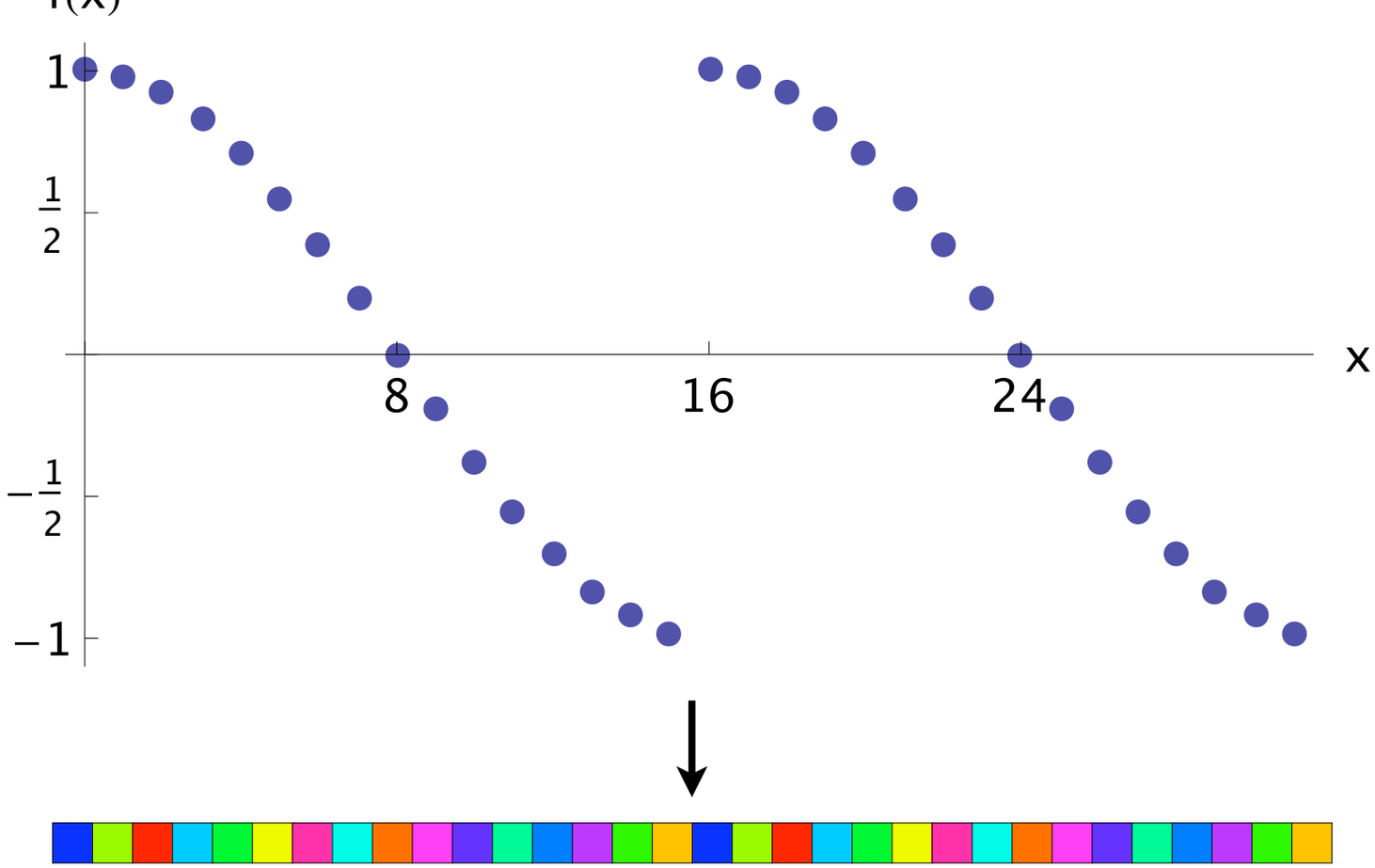


Periodic functions

A function $f : \mathbb{Z}_N \to S$ is *injectively periodic with period* $r \in \mathbb{Z}_N$ if

$$f(x) = f(y) \text{ if and only if } \frac{x-y}{r} \in \mathbb{Z}$$

Example ($N = 32$):



Producing a periodic state

Producing a periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

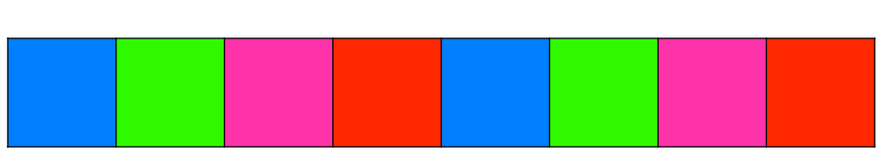
Producing a periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

Append $|0\rangle$ and compute f:

$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$$



Producing a periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

Append $|0\rangle$ and compute f:

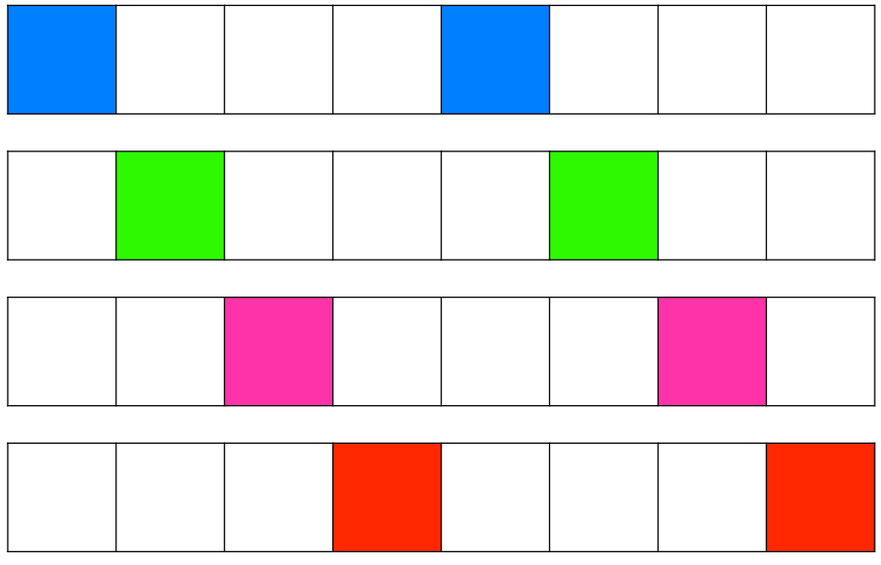
$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$$

Measure the second register:

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle$$

s uniformly random

classical value $f(s)$



Producing a periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

Append $|0\rangle$ and compute f:

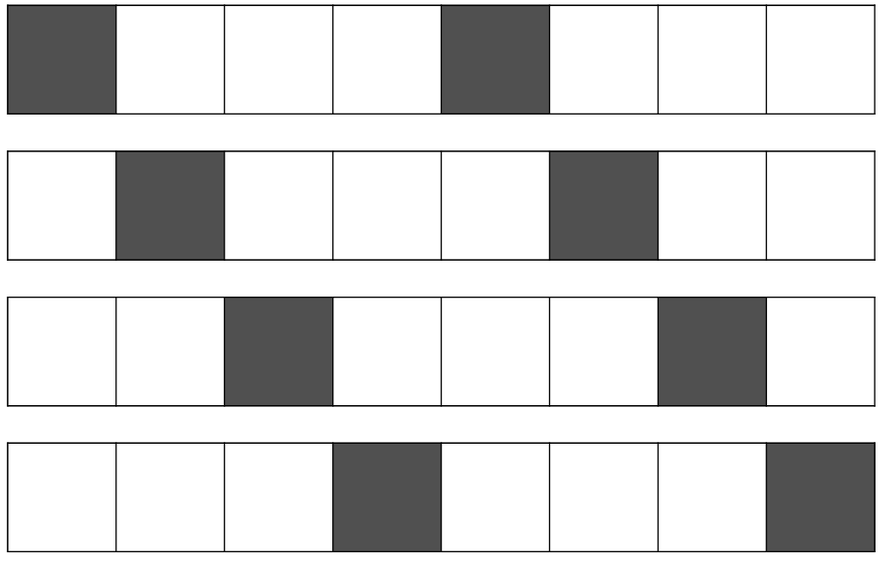
$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$$

Measure the second register:

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle$$

s uniformly random

~~classical value $f(s)$~~



Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

Apply the QFT

Recall $|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle$$

Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle \mapsto \frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle \mapsto \frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_N^{jry} |y\rangle$$

Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle \mapsto \frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_N^{jry} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_{N/r}^{jy} |y\rangle$$

A useful identity

Claim: $$\frac{1}{M}\sum_{j=0}^{M-1}\omega_M^{jk} = \delta_{k \bmod M,0}$$

A useful identity

Claim: $$\frac{1}{M}\sum_{j=0}^{M-1}\omega_M^{jk} = \delta_{k \bmod M,0}$$

Proof:

A useful identity

Claim: $$\frac{1}{M}\sum_{j=0}^{M-1}\omega_M^{jk} = \delta_{k \bmod M,0}$$

Proof: For $k = 0 \bmod M$, obvious.

A useful identity

Claim: $$\frac{1}{M}\sum_{j=0}^{M-1}\omega_M^{jk} = \delta_{k \bmod M,0}$$

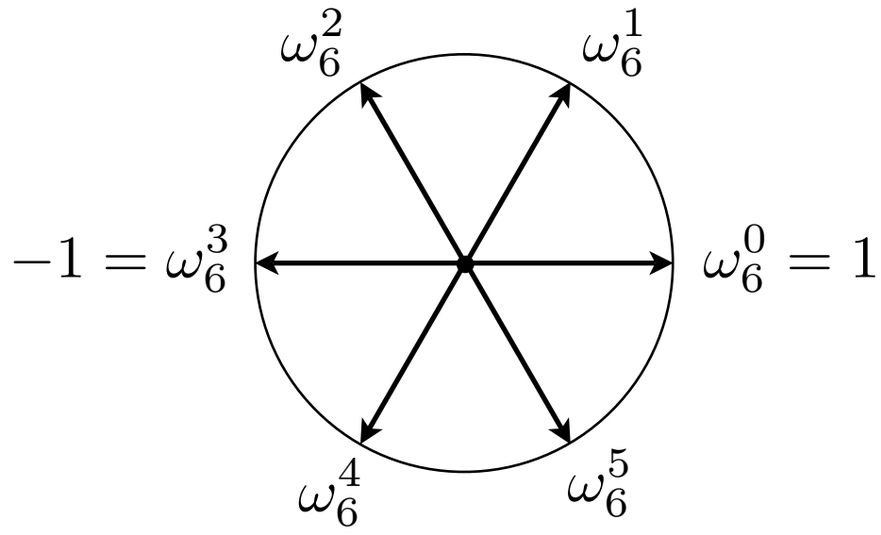
Proof: For $k = 0 \bmod M$, obvious.

For $k \neq 0$,

$$\sum_{j=0}^{M-1}\omega_M^{jk} = \frac{1-\omega_M^{kM}}{1-\omega_M^k} \quad \text{(geometric series)}$$

$$= \frac{1-e^{2\pi \mathrm{i} k}}{1-\omega_M^k}$$

$$= 0$$



Apply the QFT, continued

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle \mapsto \frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_N^{jry} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_{N/r}^{jy} |y\rangle$$

$$\frac{1}{M} \sum_{j=0}^{M-1} \omega_M^{jk} = \delta_{k \bmod M,0}$$

Apply the QFT, continued

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle \mapsto \frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_N^{jry} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_{N/r}^{jy} |y\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{y \in (N/r)\mathbb{Z}_r} \omega_N^{sy} |y\rangle$$

$$\frac{1}{M} \sum_{j=0}^{M-1} \omega_M^{jk} = \delta_{k \bmod M,0}$$

Apply the QFT, continued

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s+jr\rangle \mapsto \frac{\sqrt{r}}{N} \sum_{j=0}^{\frac{N}{r}-1} \sum_{y\in\mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y\in\mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_N^{jry} |y\rangle$$

$$= \frac{\sqrt{r}}{N} \sum_{y\in\mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{\frac{N}{r}-1} \omega_{N/r}^{jy} |y\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{y\in(N/r)\mathbb{Z}_r} \omega_N^{sy} |y\rangle$$

$$= \frac{1}{\sqrt{r}} \sum_{z\in\mathbb{Z}_r} \omega_{N/r}^{sz} |z\tfrac{N}{r}\rangle$$

$$\frac{1}{M} \sum_{j=0}^{M-1} \omega_M^{jk} = \delta_{k \bmod M,0}$$

Effect of the QFT

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle$$

periodic, period r

unknown offset s

$$\downarrow F_{\mathbb{Z}_N}$$

$$\frac{1}{\sqrt{r}} \sum_{z \in \mathbb{Z}_r} \omega_{N/r}^{sz} |z \tfrac{N}{r}\rangle$$

periodic, period N/r

zero offset

unknown phases

Learning the period

Measure $\frac{1}{\sqrt{r}} \sum_{z \in \mathbb{Z}_r} \omega_{N/r}^{sz} |z \tfrac{N}{r}\rangle$

Learning the period

Measure $\frac{1}{\sqrt{r}} \sum_{z \in \mathbb{Z}_r} \omega_{N/r}^{sz} |z \tfrac{N}{r}\rangle$

Outcome: $0, \frac{N}{r}, 2\frac{N}{r}, 3\frac{N}{r}, \ldots, N - \frac{N}{r}$ uniformly at random

Learning the period

Measure $\frac{1}{\sqrt{r}} \sum_{z \in \mathbb{Z}_r} \omega_{N/r}^{sz} |z \tfrac{N}{r}\rangle$

Outcome: $0, \frac{N}{r}, 2\frac{N}{r}, 3\frac{N}{r}, \ldots, N - \frac{N}{r}$ uniformly at random

Strategy: Compute z/r in lowest terms (Euclid's algorithm) and assume the denominator is r.

Learning the period

Measure $\frac{1}{\sqrt{r}} \sum_{z \in \mathbb{Z}_r} \omega_{N/r}^{sz} |z \tfrac{N}{r}\rangle$

Outcome: $0, \frac{N}{r}, 2\frac{N}{r}, 3\frac{N}{r}, \ldots, N - \frac{N}{r}$ uniformly at random

Strategy: Compute z/r in lowest terms (Euclid's algorithm) and assume the denominator is r.

Probability of success is

$$\frac{\varphi(r)}{r} = \Omega\left(\frac{1}{\log\log r}\right) = \Omega\left(\frac{1}{\log\log N}\right)$$

An equivalent formulation

Prepare $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$ and discard the second register:

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle$$ with s uniformly random

An equivalent formulation

Prepare $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$ and discard the second register:

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle \quad \text{with } s \text{ uniformly random}$$

$$\Updownarrow$$

$$\rho_r = \frac{r}{N^2} \sum_{s \in \mathbb{Z}_N} \sum_{j,k=0}^{\frac{N}{r}-1} |s + jr\rangle\langle s + kr| = \frac{1}{N} \sum_{s \in \mathbb{Z}_N} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle\langle s|$$

An equivalent formulation

Prepare $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$ and discard the second register:

$$\sqrt{\frac{r}{N}} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle \quad \text{with } s \text{ uniformly random}$$

$$\Updownarrow$$

$$\rho_r = \frac{r}{N^2} \sum_{s \in \mathbb{Z}_N} \sum_{j,k=0}^{\frac{N}{r}-1} |s + jr\rangle\langle s + kr| = \frac{1}{N} \sum_{s \in \mathbb{Z}_N} \sum_{j=0}^{\frac{N}{r}-1} |s + jr\rangle\langle s|$$

$$\Big\downarrow F_{\mathbb{Z}_N}$$

$$F_{\mathbb{Z}_N} \rho_r F_{\mathbb{Z}_N}^\dagger = \frac{1}{r} \sum_{z \in \mathbb{Z}} |z\tfrac{N}{r}\rangle\langle z\tfrac{N}{r}|$$

Period finding over $\mathbb{Z}$

Unknown domain of periodicity

Suppose we have a function $f : \mathbb{Z} \to S$ satisfying

$$f(x) = f(y) \text{ if and only if } \frac{x-y}{r} \in \mathbb{Z}$$

but we don't know an N for which $f(x+N) = f(x)$.

Can we still find r?

Unknown domain of periodicity

Suppose we have a function $f : \mathbb{Z} \to S$ satisfying

$$f(x) = f(y) \text{ if and only if } \frac{x-y}{r} \in \mathbb{Z}$$

but we don't know an N for which $f(x+N) = f(x)$.

Can we still find r?

Strategy: Choose a large N, and hope that the procedure still works, even though r will probably not divide r.

Producing an (almost) periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

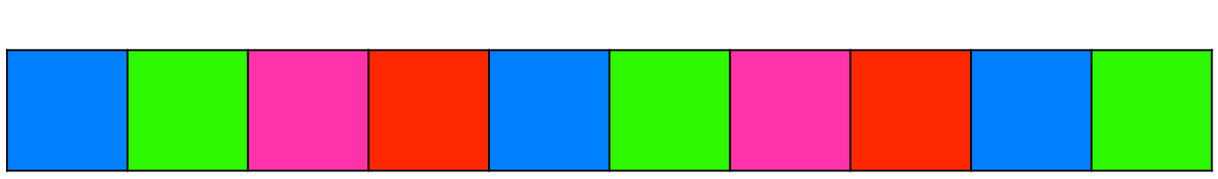
Producing an (almost) periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

Append $|0\rangle$ and compute f:

$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$$



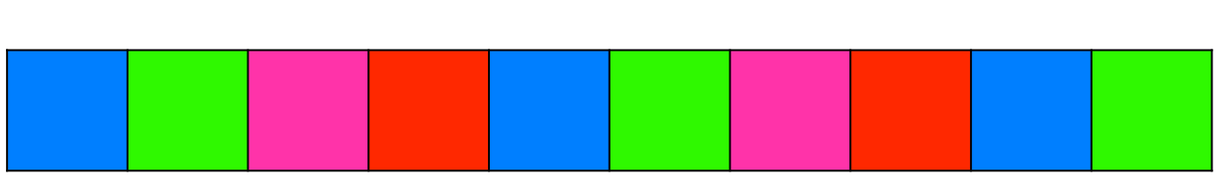
Producing an (almost) periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

Append $|0\rangle$ and compute f:

$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$$

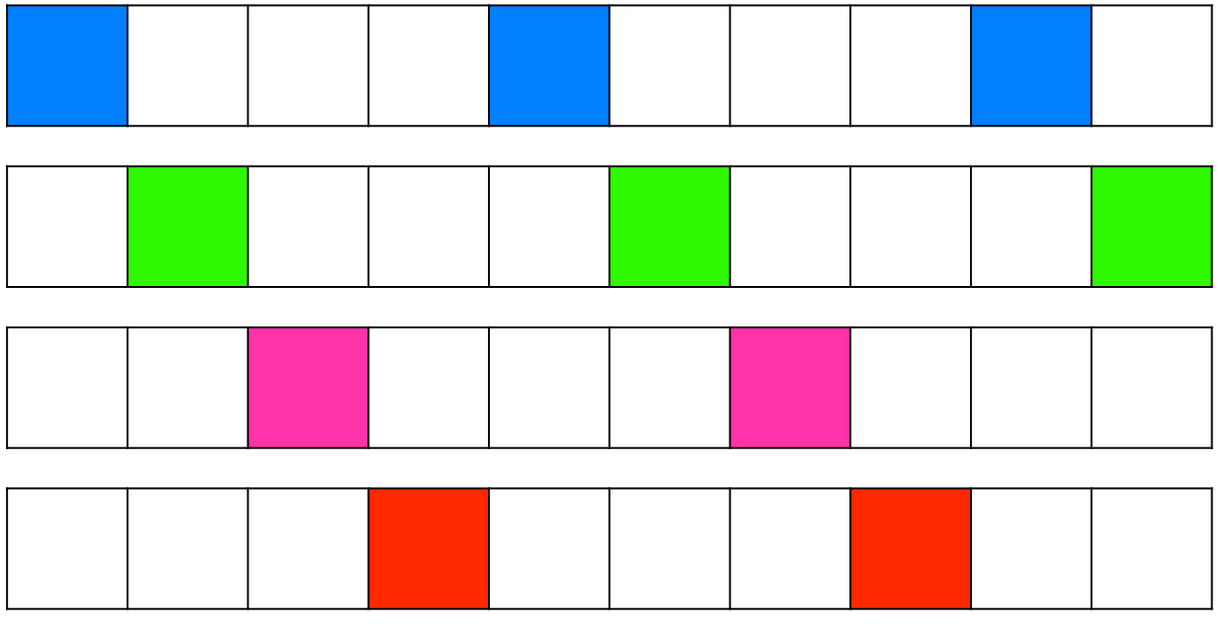


Measure the second register:

$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |s + jr\rangle$$

$n \approx N/r$

$s \approx$ uniformly random



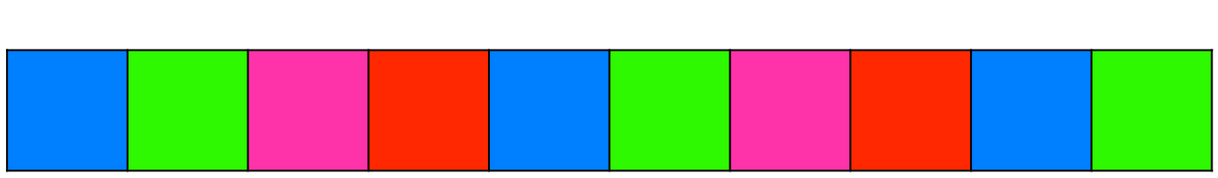
Producing an (almost) periodic state

Create the state

$$|\mathbb{Z}_N\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$$

Append $|0\rangle$ and compute f:

$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x, f(x)\rangle$$

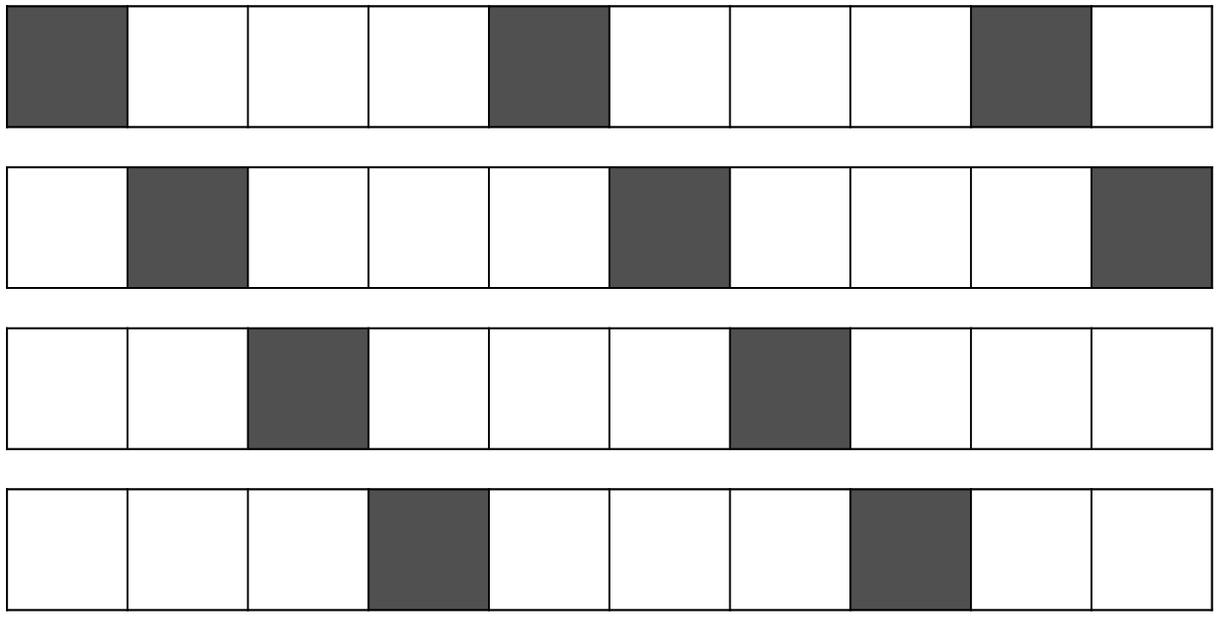


Measure the second register:

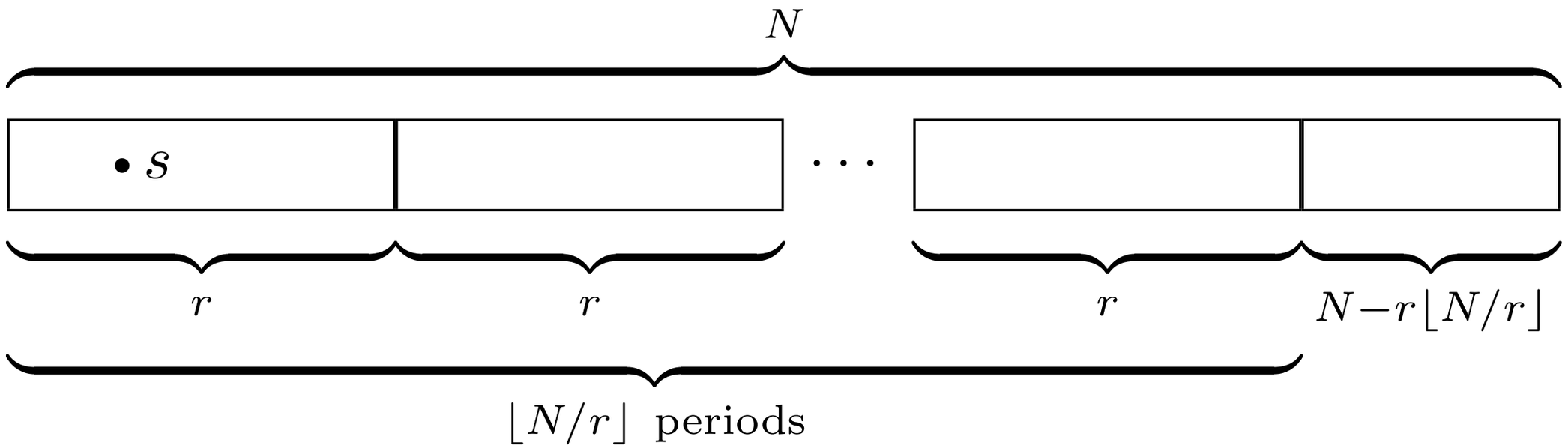
$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |s + jr\rangle$$

$n \approx N/r$

$s \approx$ uniformly random



The (almost) periodic state



$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |s + jr\rangle \qquad n = \begin{cases} \lfloor N/r \rfloor + 1 & s < N - r\lfloor N/r \rfloor \\ \lfloor N/r \rfloor & \text{otherwise.} \end{cases}$$

where s occurs with probability n/N

Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

Apply the QFT

Recall $|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$

$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |s + jr\rangle$$

Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |s + jr\rangle \mapsto \frac{1}{\sqrt{nN}} \sum_{j=0}^{n-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |s + jr\rangle \mapsto \frac{1}{\sqrt{nN}} \sum_{j=0}^{n-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

$$= \frac{1}{\sqrt{nN}} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{n-1} \omega_N^{jry} |y\rangle$$

Apply the QFT

Recall $$|x\rangle \overset{F_{\mathbb{Z}_N}}{\longmapsto} \frac{1}{\sqrt{N}} \sum_{y \in \mathbb{Z}_N} \omega_N^{xy} |y\rangle$$

$$\frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} |s + jr\rangle \mapsto \frac{1}{\sqrt{nN}} \sum_{j=0}^{n-1} \sum_{y \in \mathbb{Z}_N} \omega_N^{(s+jr)y} |y\rangle$$

$$= \frac{1}{\sqrt{nN}} \sum_{y \in \mathbb{Z}_N} \omega_N^{sy} \sum_{j=0}^{n-1} \omega_N^{jry} |y\rangle$$

Probability of observing y: $\Pr(y) = \frac{1}{nN} \left| \sum_{j=0}^{n-1} \omega_N^{jry} \right|^2$

The distribution of measurement outcomes

Probability of observing y: $\Pr(y) = \frac{1}{nN}\left|\sum_{j=0}^{n-1}\omega_N^{jry}\right|^2$

The distribution of measurement outcomes

Probability of observing y: $\Pr(y) = \frac{1}{nN}\left|\sum_{j=0}^{n-1}\omega_N^{jry}\right|^2$

When r divides N, we have already seen that $\Pr(y) = \frac{1}{r}\delta_{y \bmod \frac{N}{r},0}$

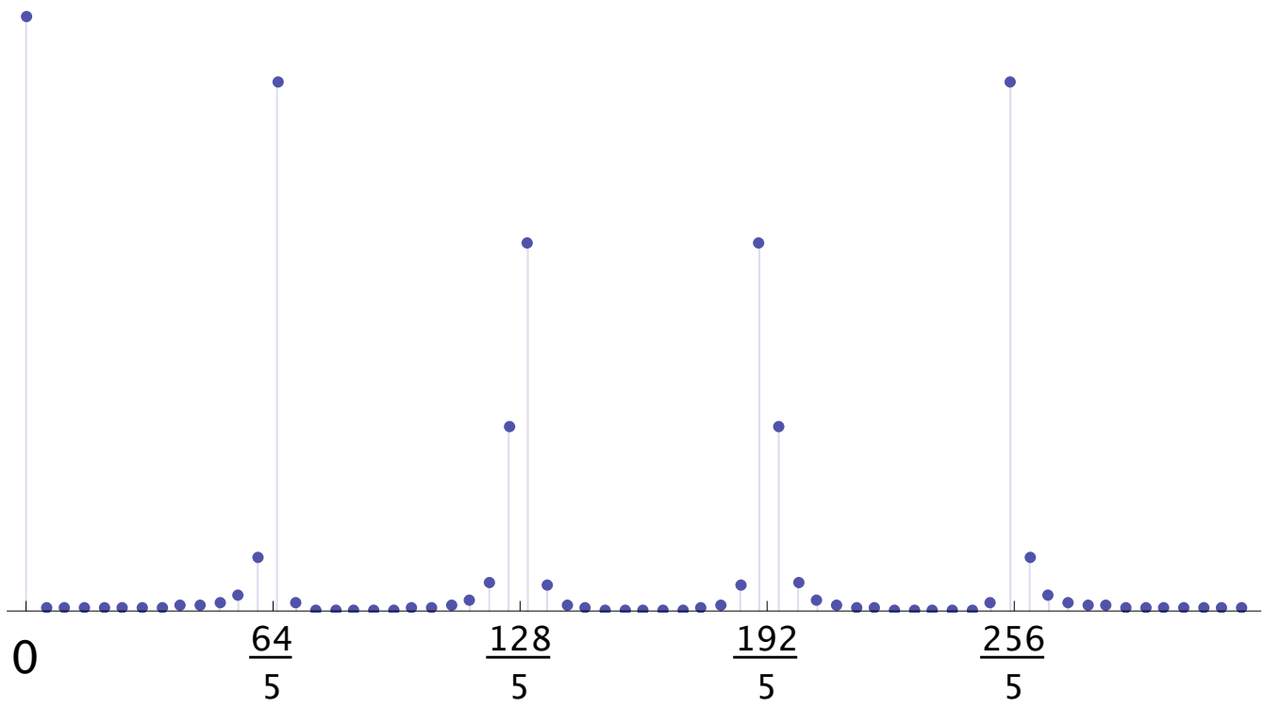
The distribution of measurement outcomes

Probability of observing y: $\Pr(y) = \frac{1}{nN} \left| \sum_{j=0}^{n-1} \omega_N^{jry} \right|^2$

When r divides N, we have already seen that $\Pr(y) = \frac{1}{r} \delta_{y \bmod \frac{N}{r}, 0}$

In general, $\Pr(y)$ is sharply peaked around multiples of N/r.

Example ($N = 64$, $r = 5$):



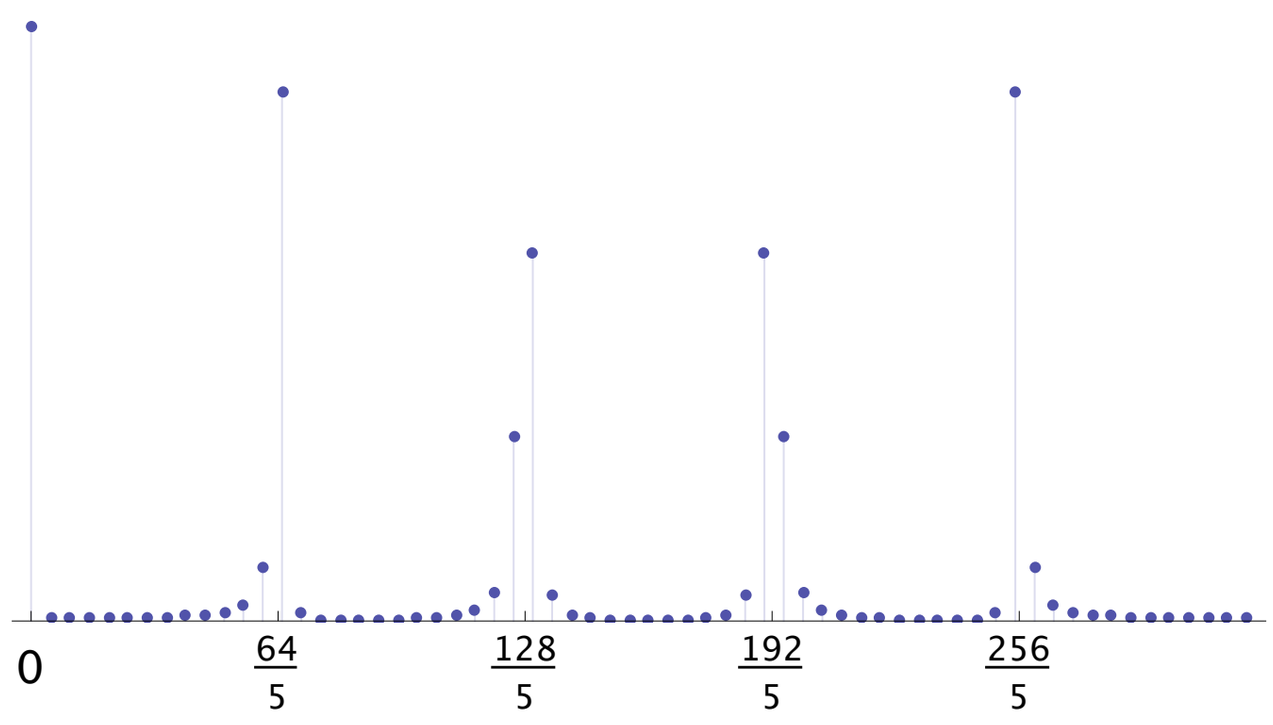
The distribution of measurement outcomes

Probability of observing y: $\Pr(y) = \frac{1}{nN} \left| \sum_{j=0}^{n-1} \omega_N^{jry} \right|^2$

When r divides N, we have already seen that $\Pr(y) = \frac{1}{r} \delta_{y \bmod \frac{N}{r}, 0}$

In general, $\Pr(y)$ is sharply peaked around multiples of N/r.

Example ($N = 64$, $r = 5$):



Exercise: Show $\Pr\left(y \in \{0, \lfloor \frac{N}{r} \rceil, \lfloor 2\frac{N}{r} \rceil, \ldots, \lfloor (r-1)\frac{N}{r} \rceil\}\right) = \Omega(1)$.

Continued fractions

Problem: Given a sample from $\{0, \lfloor \frac{N}{r} \rceil, \lfloor 2\frac{N}{r} \rceil, \ldots, \lfloor (r-1)\frac{N}{r} \rceil\}$, determine r.

Continued fractions

Problem: Given a sample from $\{0, \lfloor \frac{N}{r} \rceil, \lfloor 2\frac{N}{r} \rceil, \ldots, \lfloor (r-1)\frac{N}{r} \rceil\}$, determine r.

Compute the *continued fraction expansion*

$$\frac{\lfloor jN/r \rceil}{N} = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

Continued fractions

Problem: Given a sample from $\{0, \lfloor \frac{N}{r} \rceil, \lfloor 2\frac{N}{r} \rceil, \ldots, \lfloor (r-1)\frac{N}{r} \rceil\}$, determine r.

Compute the *continued fraction expansion*

$$\frac{\lfloor jN/r \rceil}{N} = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

Suppose we know that $r < r_{\max}$. Then the closest continued fraction approximation with denominator less than $r_{\max}$ is guaranteed to be r provided $N > (r_{\max})^2$.

Continued fractions

Problem: Given a sample from $\{0, \lfloor \frac{N}{r} \rceil, \lfloor 2\frac{N}{r} \rceil, \ldots, \lfloor (r-1)\frac{N}{r} \rceil\}$, determine r.

Compute the *continued fraction expansion*

$$\frac{\lfloor jN/r \rceil}{N} = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

Suppose we know that $r < r_{\max}$. Then the closest continued fraction approximation with denominator less than $r_{\max}$ is guaranteed to be r provided $N > (r_{\max})^2$.

If we are not given an a priori upper bound $r_{\max}$, we can start with $r_{\max} = 2$ and successively double it until we find r; then the running time will be $\mathrm{poly}(\log r)$.

Application: Order finding

Let G be a group. The *order* of $g \in G$ is the smallest $r \in \{1, 2, 3, ...\}$ such that $g^r = 1$.

Application: Order finding

Let G be a group. The *order* of $g \in G$ is the smallest $r \in \{1, 2, 3, ...\}$ such that $g^r = 1$.

The function $f : \mathbb{Z} \to G$ defined by $f(x) = g^x$ is (injectively) periodic with period equal to the order of g in G.

Application: Order finding

Let G be a group. The *order* of $g \in G$ is the smallest $r \in \{1, 2, 3, ...\}$ such that $g^r = 1$.

The function $f : \mathbb{Z} \to G$ defined by $f(x) = g^x$ is (injectively) periodic with period equal to the order of g in G.

periodicity: $f(x + r) = g^{x+r} = g^x = f(x)$

Application: Order finding

Let G be a group. The *order* of $g \in G$ is the smallest $r \in \{1, 2, 3, ...\}$ such that $g^r = 1$.

The function $f : \mathbb{Z} \to G$ defined by $f(x) = g^x$ is (injectively) periodic with period equal to the order of g in G.

periodicity: $f(x + r) = g^{x+r} = g^x = f(x)$

injectivity: there is no $x < r$ for which $f(x) = f(0)$

Application: Order finding

Let G be a group. The *order* of $g \in G$ is the smallest $r \in \{1, 2, 3, ...\}$ such that $g^r = 1$.

The function $f : \mathbb{Z} \to G$ defined by $f(x) = g^x$ is (injectively) periodic with period equal to the order of g in G.

periodicity: $f(x + r) = g^{x+r} = g^x = f(x)$

injectivity: there is no $x < r$ for which $f(x) = f(0)$

So there is an efficient (running time $\mathrm{poly}(\log |G|)$) quantum algorithm for order finding.

Reduction of factoring to period finding

Factoring → Finding a nontrivial factor

Suppose we want to factor the positive integer N.

Since primality can be tested efficiently, it suffices to give a procedure for finding a nontrivial factor of N with constant probability.

Factoring → Finding a nontrivial factor

Suppose we want to factor the positive integer N.

Since primality can be tested efficiently, it suffices to give a procedure for finding a nontrivial factor of N with constant probability.

```
function factor(N)
    if N is prime
        output N
    else
        repeat
            x=find_nontrivial_factor(N)
        until success
        factor(x)
        factor(N/x)
    end if
```

Factoring → Finding a nontrivial factor

Suppose we want to factor the positive integer N.

Since primality can be tested efficiently, it suffices to give a procedure for finding a nontrivial factor of N with constant probability.

```
function factor(N)
    if N is prime
        output N
    else
        repeat
            x=find_nontrivial_factor(N)
        until success
        factor(x)
        factor(N/x)
    end if
```

We can assume N is odd, since it is easy to find the factor 2.

Factoring → Finding a nontrivial factor

Suppose we want to factor the positive integer N.

Since primality can be tested efficiently, it suffices to give a procedure for finding a nontrivial factor of N with constant probability.

```
function factor(N)
   if N is prime
      output N
   else
      repeat
         x=find_nontrivial_factor(N)
      until success
      factor(x)
      factor(N/x)
   end if
```

We can assume N is odd, since it is easy to find the factor 2.

We can also assume that N contains at least two distinct prime powers, since it is easy to check if it is a power of some integer.

Miller's reduction

Factoring N reduces to order finding in $\mathbb{Z}_N^\times$ (Miller 1976).

Miller's reduction

Factoring N reduces to order finding in $\mathbb{Z}_N^\times$ (Miller 1976).

Choose $a \in \{2, 3, \ldots, N-1\}$ uniformly at random.
If $\gcd(a, N) \neq 1$, then it is a nontrivial factor of N; otherwise $a \in \mathbb{Z}_N^\times$.

Miller's reduction

Factoring N reduces to order finding in $\mathbb{Z}_N^\times$ (Miller 1976).

Choose $a \in \{2, 3, \ldots, N-1\}$ uniformly at random.
If $\gcd(a, N) \neq 1$, then it is a nontrivial factor of N; otherwise $a \in \mathbb{Z}_N^\times$.

Let r denote the (multiplicative) order of a modulo N.

Miller's reduction

Factoring N reduces to order finding in $\mathbb{Z}_N^\times$ (Miller 1976).

Choose $a \in \{2, 3, \ldots, N-1\}$ uniformly at random.
If $\gcd(a, N) \neq 1$, then it is a nontrivial factor of N; otherwise $a \in \mathbb{Z}_N^\times$.

Let r denote the (multiplicative) order of a modulo N.

Suppose r is even. Then

$$a^r = 1 \bmod N$$
$$\Updownarrow$$
$$(a^{r/2})^2 - 1 = 0 \bmod N$$

Miller's reduction

Factoring N reduces to order finding in $\mathbb{Z}_N^\times$ (Miller 1976).

Choose $a \in \{2, 3, \ldots, N-1\}$ uniformly at random.
If $\gcd(a, N) \neq 1$, then it is a nontrivial factor of N; otherwise $a \in \mathbb{Z}_N^\times$.

Let r denote the (multiplicative) order of a modulo N.

Suppose r is even. Then

$$a^r = 1 \bmod N$$
$$\Updownarrow$$
$$(a^{r/2})^2 - 1 = 0 \bmod N$$
$$\Updownarrow$$
$$(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N$$

Miller's reduction

Factoring N reduces to order finding in $\mathbb{Z}_N^\times$ (Miller 1976).

Choose $a \in \{2, 3, \ldots, N-1\}$ uniformly at random.
If $\gcd(a, N) \neq 1$, then it is a nontrivial factor of N; otherwise $a \in \mathbb{Z}_N^\times$.

Let r denote the (multiplicative) order of a modulo N.

Suppose r is even. Then

$$a^r = 1 \bmod N$$
$$\Updownarrow$$
$$(a^{r/2})^2 - 1 = 0 \bmod N$$
$$\Updownarrow$$
$$(a^{r/2} - 1)(a^{r/2} + 1) = 0 \bmod N$$

so we might hope for $\gcd(\mathrm{a}^{r/2} - 1, N)$ to be a nontrivial factor of N.

Miller's reduction

Given $(a^{r/2}-1)(a^{r/2}+1) = 0 \bmod N$

when does $\gcd(a^{r/2}-1, N)$ give a nontrivial factor of N?

Miller's reduction

Given $(a^{r/2}-1)(a^{r/2}+1) = 0 \bmod N$

when does $\gcd(a^{r/2}-1, N)$ give a nontrivial factor of N?

Note that $a^{r/2}-1 \neq 0 \bmod N$ (otherwise the order of a would be $r/2$, or smaller).

Miller's reduction

Given $(a^{r/2}-1)(a^{r/2}+1) = 0 \bmod N$

when does $\gcd(a^{r/2}-1, N)$ give a nontrivial factor of N?

Note that $a^{r/2}-1 \neq 0 \bmod N$ (otherwise the order of a would be $r/2$, or smaller).

So it suffices to ensure that $a^{r/2}+1 \neq 0 \bmod N$.

Miller's reduction

Given $(a^{r/2}-1)(a^{r/2}+1) = 0 \bmod N$

when does $\gcd(a^{r/2}-1, N)$ give a nontrivial factor of N?

Note that $a^{r/2} - 1 \neq 0 \bmod N$ (otherwise the order of a would be $r/2$, or smaller).

So it suffices to ensure that $a^{r/2} + 1 \neq 0 \bmod N$.

Lemma: Suppose $a \in \mathbb{Z}_N^\times$ is chosen uniformly at random, where N is an odd integer with at least two distinct prime factors. Then with probability at least $1/2$, the multiplicative order r of a is even and $a^{r/2} \neq -1 \bmod N$.

Proof (part 1 of 2)

Let $N = p_1^{m_1} \times \cdots \times p_k^{m_k}$ (p_i distinct primes, $k \geq 2$)

$a = a_i \bmod p_i^{m_i}$

$r_i =$ order of a_i modulo $p_i^{m_i}$

$2^{c_i} =$ largest power of 2 that divides r_i

Proof (part 1 of 2)

Let $N = p_1^{m_1} \times \cdots \times p_k^{m_k}$ (p_i distinct primes, $k \geq 2$)

$a = a_i \bmod p_i^{m_i}$

$r_i =$ order of a_i modulo $p_i^{m_i}$

$2^{c_i} =$ largest power of 2 that divides r_i

Claim 1: If r is odd or $a^{r/2} + 1 \neq 0 \bmod N$, then $c_1 = \cdots = c_k$.

Proof (part 1 of 2)

Let $N = p_1^{m_1} \times \cdots \times p_k^{m_k}$ (p_i distinct primes, $k \geq 2$)

$a = a_i \bmod p_i^{m_i}$

$r_i =$ order of a_i modulo $p_i^{m_i}$

$2^{c_i} =$ largest power of 2 that divides r_i

Claim 1: If r is odd or $a^{r/2} + 1 \neq 0 \bmod N$, then $c_1 = \cdots = c_k$.

Since $r = \mathrm{lcm}(r_1, \ldots, r_k)$, r is odd iff $c_1 = \cdots = c_k = 0$.

Proof (part 1 of 2)

Let $N = p_1^{m_1} \times \cdots \times p_k^{m_k}$ (p_i distinct primes, $k \geq 2$)

$a = a_i \bmod p_i^{m_i}$

$r_i =$ order of a_i modulo $p_i^{m_i}$

$2^{c_i} =$ largest power of 2 that divides r_i

Claim 1: If r is odd or $a^{r/2} + 1 \neq 0 \bmod N$, then $c_1 = \cdots = c_k$.

Since $r = \mathrm{lcm}(r_1, \ldots, r_k)$, r is odd iff $c_1 = \cdots = c_k = 0$.

If r is even and $a^{r/2} = -1 \bmod N$,

then $a^{r/2} = -1 \bmod p_i^{m_i}$ for each i,

so r_i does not divide $r/2$; but notice that r_i does divide r.

Proof (part 1 of 2)

Let $N = p_1^{m_1} \times \cdots \times p_k^{m_k}$ (p_i distinct primes, $k \geq 2$)

$a = a_i \bmod p_i^{m_i}$

$r_i =$ order of a_i modulo $p_i^{m_i}$

$2^{c_i} =$ largest power of 2 that divides r_i

Claim 1: If r is odd or $a^{r/2} + 1 \neq 0 \bmod N$, then $c_1 = \cdots = c_k$.

Since $r = \operatorname{lcm}(r_1, \ldots, r_k)$, r is odd iff $c_1 = \cdots = c_k = 0$.

If r is even and $a^{r/2} = -1 \bmod N$,
then $a^{r/2} = -1 \bmod p_i^{m_i}$ for each i,
so r_i does not divide $r/2$; but notice that r_i does divide r.

Hence r/r_i is an odd integer for each i, and every r_i must contain the same number of powers of 2 as r.

Proof (part 2 of 2)

Claim 2: $\text{Prob}(c_i = \text{any particular value}) \leq 1/2$

Proof (part 2 of 2)

Claim 2: $\mathrm{Prob}(c_i = \text{any particular value}) \leq 1/2$

(Then the lemma follows, because in particular $\mathrm{Prob}(c_1 = c_2) \leq 1/2$.)

Proof (part 2 of 2)

Claim 2: $\mathrm{Prob}(c_i = \text{any particular value}) \leq 1/2$

(Then the lemma follows, because in particular $\mathrm{Prob}(c_1 = c_2) \leq 1/2$.)

$$a \in \mathbb{Z}_N^\times \quad \Leftrightarrow \quad a_i \in \mathbb{Z}_{p_i^{m_i}}^\times$$

uniformly at random ⇔ uniformly at random

Proof (part 2 of 2)

Claim 2: $\mathrm{Prob}(c_i = \text{any particular value}) \leq 1/2$

(Then the lemma follows, because in particular $\mathrm{Prob}(c_1 = c_2) \leq 1/2$.)

$$a \in \mathbb{Z}_N^\times \quad \Leftrightarrow \quad a_i \in \mathbb{Z}_{p_i^{m_i}}^\times$$

uniformly at random ⇔ uniformly at random

Since $\mathbb{Z}_{p_i^{m_i}}^\times$ is cyclic and of even order, exactly half its elements have the maximal value of c_i, so in particular the probability of any particular c_i is at most $1/2$.

Shor's factoring algorithm

Input: Integer N Output: A nontrivial factor of N

1. Choose a random $a \in \{2, 3, \ldots, N-1\}$
2. Compute $\gcd(a, N)$; if it is not 1 then it is a nontrivial factor, and otherwise we continue
3. Prepare the uniform superposition $|\mathbb{Z}_{N^2}\rangle$ (or replace N^2 by the next largest power of 2)
4. Append an ancilla register, and conditioned on the value x in the first register, compute $f(x) = a^x \bmod N$ in the ancilla register; discard the ancilla
5. Perform the quantum Fourier transform over $\mathbb{Z}_{N^2}$
6. Measure in the computational basis
7. Compute the continued fraction expansion of the result divided by N^2, obtaining the best approximation with denominator less than N; call this denominator r
8. Compute $\gcd(a^{r/2} - 1, N)$. If it is 1 or N then we have failed, and we start over; otherwise the result is a nontrivial factor of N.

Performance

Most expensive step: Modular exponentiation.
Using repeated squaring, this can be done in time $O((\log N)^3)$.

$\Rightarrow$ Running time of Shor's algorithm is $O((\log N)^3)$

In contrast, the best known classical algorithm for factoring N (the *number field sieve*) takes time $2^{O((\log N)^{1/3}(\log\log N)^{2/3})}$.

Performance

Most expensive step: Modular exponentiation.
Using repeated squaring, this can be done in time $O((\log N)^3)$.

$\Rightarrow$ Running time of Shor's algorithm is $O((\log N)^3)$

In contrast, the best known classical algorithm for factoring N (the *number field sieve*) takes time $2^{O((\log N)^{1/3}(\log\log N)^{2/3})}$.

Note that the quantum part of Shor's algorithm can be highly parallelized: with polynomial-time classical pre- and post-processing, can achieve depth $O((\log\log N)^2)$ and size $O((\log N)^3)$ (Cleve and Watrous 2000).

Beyond factoring

So much more than Shor and Grover

- Quantum simulation
- Algebraic problems

  Factoring, discrete log, decomposing abelian groups, Pell's equation, principal ideal problem, computing unit/class groups of number fields, shifted Legendre symbol, approximating Gauss sums, counting points on curves, some nonabelian hidden subgroup problems, ...

- Search and its applications (Peter Høyer's lectures yesterday)

  Unstructured search (decision, finding, counting), collision, median finding, graph connectivity, minimum spanning trees, single source shortest paths, matchings, network flows, ...

- Quantum walk algorithms (Eddie Farhi's lectures on Friday)

  Black box graph traversal, spatial search, element distinctness, triangle finding, checking matrix multiplication, testing group commutativity, formula evaluation, ...

- Approximation of #P-hard problems

  Jones polynomial, HOMFLYPT polynomial, Tutte polynomial/Potts model partition function, ...

- Miscellanea

  Oracle interrogation, ordered search, gradient estimation, ...

Discrete log

Let $G = \{g^\alpha : \alpha \in \mathbb{Z}_N\}$ be a cyclic group generated by g.

Given $x \in G$, define $\log_g x := \min\{\ell \in \mathbb{Z}^+ : g^\ell = x\}$.

Discrete log

Let $G = \{g^\alpha : \alpha \in \mathbb{Z}_N\}$ be a cyclic group generated by g.

Given $x \in G$, define $\log_g x := \min\{\ell \in \mathbb{Z}^+ : g^\ell = x\}$.

Computing $\log_g x$ is (apparently) classically hard, and (as for factoring) this assumption is used in cryptographic protocols.

Common groups:

	best classical algorithm
$G = \mathbb{Z}_M^\times$	$2^{O((\log M)^{1/3}(\log\log M)^{2/3})}$
$G =$ elliptic curve	$O(\sqrt{N}) = 2^{O(\log N)}$

Discrete log

Let $G = \{g^\alpha : \alpha \in \mathbb{Z}_N\}$ be a cyclic group generated by g.

Given $x \in G$, define $\log_g x := \min\{\ell \in \mathbb{Z}^+ : g^\ell = x\}$.

Computing $\log_g x$ is (apparently) classically hard, and (as for factoring) this assumption is used in cryptographic protocols.

Common groups:

	best classical algorithm
$G = \mathbb{Z}_M^\times$	$2^{O((\log M)^{1/3}(\log\log M)^{2/3})}$
$G =$ elliptic curve	$O(\sqrt{N}) = 2^{O(\log N)}$

But there is an efficient quantum algorithm for discrete log in general groups (Shor 1994).

The abelian hidden subgroup problem

Let G be an abelian group.

We say that $f : G \to S$ *hides* the subgroup $H \leq G$ if

$$f(x) = f(y) \text{ if and only if } x - y \in H$$

Abelian HSP: Given the ability to query f, find a generating set for H.

The abelian hidden subgroup problem

Let G be an abelian group.

We say that $f : G \to S$ *hides* the subgroup $H \leq G$ if

$$f(x) = f(y) \text{ if and only if } x - y \in H$$

Abelian HSP: Given the ability to query f, find a generating set for H.

Example: If $G = \mathbb{Z}_N$, then f hides a subgroup isomorphic to $\mathbb{Z}_{N/r}$ iff it is injectively periodic with period r

The abelian hidden subgroup problem

Let G be an abelian group.

We say that $f : G \to S$ *hides* the subgroup $H \leq G$ if

$$f(x) = f(y) \text{ if and only if } x - y \in H$$

Abelian HSP: Given the ability to query f, find a generating set for H.

Example: If $G = \mathbb{Z}_N$, then f hides a subgroup isomorphic to $\mathbb{Z}_{N/r}$ iff it is injectively periodic with period r

Example: The function $f(\alpha, \beta) = x^\alpha g^\beta$ hides $\langle (\alpha, \alpha \log_g x) : \alpha \in \mathbb{Z}_N \rangle \leq \mathbb{Z}_N \times \mathbb{Z}_N$

The abelian hidden subgroup problem

Let G be an abelian group.

We say that $f : G \to S$ *hides* the subgroup $H \leq G$ if

$$f(x) = f(y) \text{ if and only if } x - y \in H$$

Abelian HSP: Given the ability to query f, find a generating set for H.

Example: If $G = \mathbb{Z}_N$, then f hides a subgroup isomorphic to $\mathbb{Z}_{N/r}$ iff it is injectively periodic with period r

Example: The function $f(\alpha, \beta) = x^\alpha g^\beta$ hides $\langle (\alpha, \alpha \log_g x) : \alpha \in \mathbb{Z}_N \rangle \leq \mathbb{Z}_N \times \mathbb{Z}_N$

There is an efficient quantum algorithm for the hidden subgroup problem in any finitely generated abelian group.

Counting points on curves

Problem: How many solutions are there to the polynomial equation

$$f(x, y) = 0$$

where $x, y \in \mathbb{Z}_p$ (or more generally, a finite field)?

Counting points on curves

Problem: How many solutions are there to the polynomial equation

$$f(x, y) = 0$$

where $x, y \in \mathbb{Z}_p$ (or more generally, a finite field)?

This problem appears to be hard for a classical computer.

Counting points on curves

Problem: How many solutions are there to the polynomial equation

$$f(x, y) = 0$$

where $x, y \in \mathbb{Z}_p$ (or more generally, a finite field)?

This problem appears to be hard for a classical computer.

But Kedlaya showed it can be solved in time $\mathrm{poly}(\log p, d)$ on a quantum computer, where d is the degree of the polynomial.

Counting points on curves

Problem: How many solutions are there to the polynomial equation

$$f(x, y) = 0$$

where $x, y \in \mathbb{Z}_p$ (or more generally, a finite field)?

This problem appears to be hard for a classical computer.

But Kedlaya showed it can be solved in time $\mathrm{poly}(\log p, d)$ on a quantum computer, where d is the degree of the polynomial.

Approach: Use the algorithm for the abelian HSP to learn the structure of the *class group* of the curve, which determines the number of points.

Pell's equation and number fields

Pell's equation: Given a squarefree positive integer d, find integer solutions (x, y) to $x^2 - dy^2 = 1$.

Pell's equation and number fields

Pell's equation: Given a squarefree positive integer d, find integer solutions (x, y) to $x^2 - dy^2 = 1$.

There are infinitely many solutions, but they are all generated by one *fundamental solution* (x_1, y_1).

Pell's equation and number fields

Pell's equation: Given a squarefree positive integer d, find integer solutions (x, y) to $x^2 - dy^2 = 1$.

There are infinitely many solutions, but they are all generated by one *fundamental solution* (x_1, y_1).

Even the fundamental solution may be too large just to write down in polynomial time, but it can be encoded in the integer part of the *regulator*, $\log(x_1 + y_1\sqrt{d})$.

Pell's equation and number fields

Pell's equation: Given a squarefree positive integer d, find integer solutions (x, y) to $x^2 - dy^2 = 1$.

There are infinitely many solutions, but they are all generated by one *fundamental solution* (x_1, y_1).

Even the fundamental solution may be too large just to write down in polynomial time, but it can be encoded in the integer part of the *regulator*, $\log(x_1 + y_1\sqrt{d})$.

This can be found efficiently by a quantum computer using a generalization of period finding to the real numbers.

Pell's equation and number fields

Pell's equation: Given a squarefree positive integer d, find integer solutions (x, y) to $x^2 - dy^2 = 1$.

There are infinitely many solutions, but they are all generated by one *fundamental solution* (x_1, y_1).

Even the fundamental solution may be too large just to write down in polynomial time, but it can be encoded in the integer part of the *regulator*, $\log(x_1 + y_1\sqrt{d})$.

This can be found efficiently by a quantum computer using a generalization of period finding to the real numbers.

Similar ideas lead to efficient quantum algorithms for computing properties of number fields.

The nonabelian hidden subgroup problem

Let G be any finite group.

We say that $f : G \to S$ *hides* the subgroup $H \leq G$ if

$$f(x) = f(y) \text{ if and only if } xy^{-1} \in H$$

HSP: Given the ability to query f, find a generating set for H.

The nonabelian hidden subgroup problem

Let G be any finite group.

We say that $f : G \to S$ *hides* the subgroup $H \leq G$ if

$$f(x) = f(y) \text{ if and only if } xy^{-1} \in H$$

HSP: Given the ability to query f, find a generating set for H.

When G is nonabelian, this can be significantly harder.

The nonabelian hidden subgroup problem

Let G be any finite group.

We say that $f : G \to S$ *hides* the subgroup $H \leq G$ if

$$f(x) = f(y) \text{ if and only if } xy^{-1} \in H$$

HSP: Given the ability to query f, find a generating set for H.

When G is nonabelian, this can be significantly harder.

Potential applications:

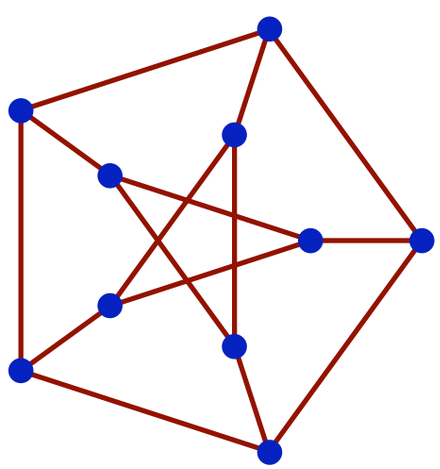
- Graph isomorphism
- Lattice problems

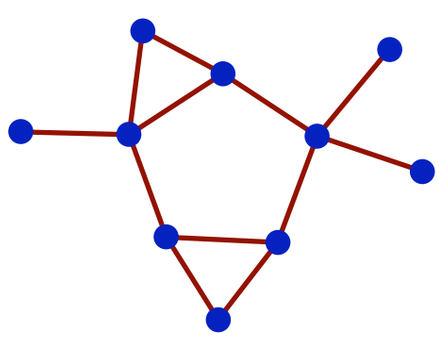
Graph automorphism

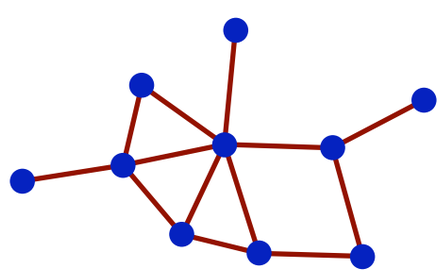
Problem: Given an n-vertex graph Γ, what is its automorphism group?

Graph automorphism

Problem: Given an n-vertex graph Γ, what is its automorphism group?

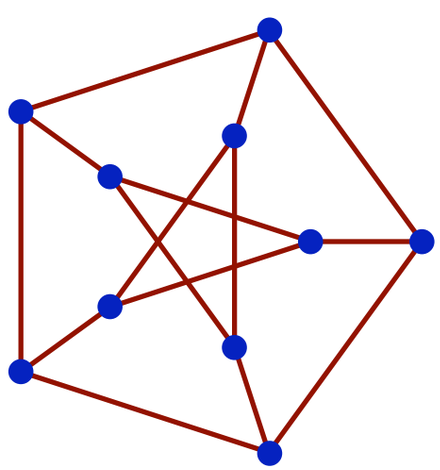
$\Gamma_1 =$ 

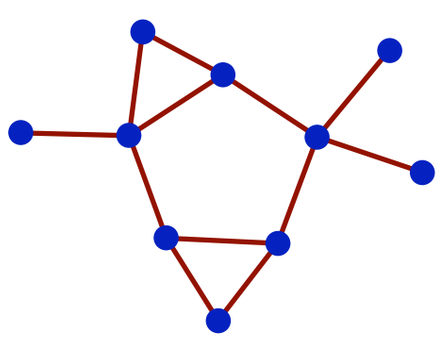
$\Gamma_2 =$ 

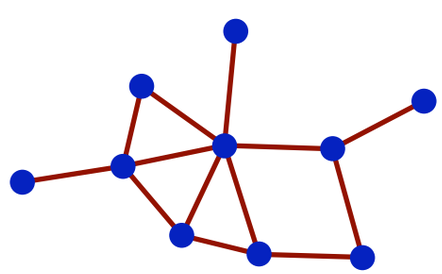
$\Gamma_3 =$ 

Graph automorphism

Problem: Given an n-vertex graph Γ, what is its automorphism group?

$\Gamma_1 =$ 

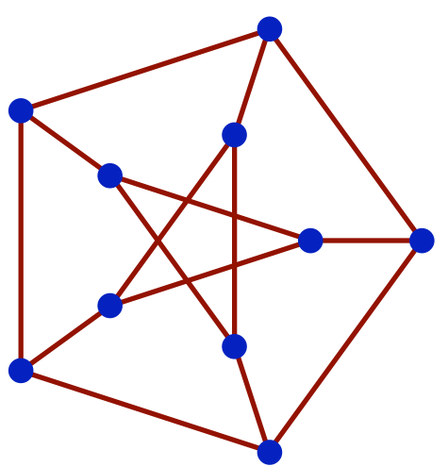
$\Gamma_2 =$ 

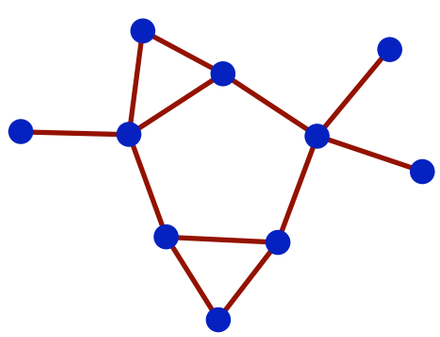
$\Gamma_3 =$ 

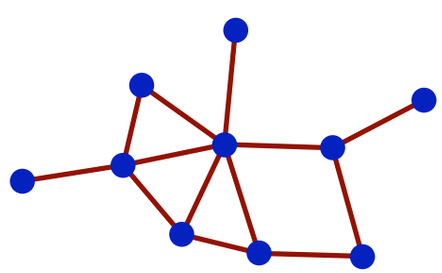
$\mathrm{Aut}(\Gamma_1) \cong S_5$

Graph automorphism

Problem: Given an n-vertex graph Γ, what is its automorphism group?

$\Gamma_1 =$ 

$\Gamma_2 =$ 

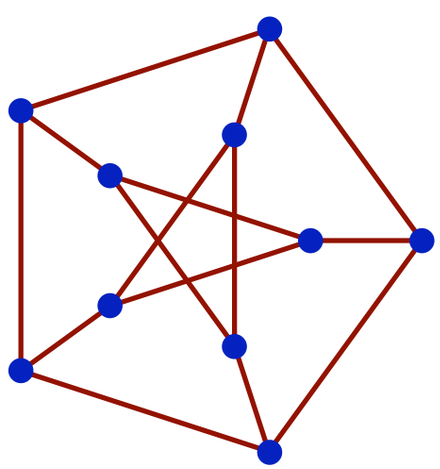
$\Gamma_3 =$ 

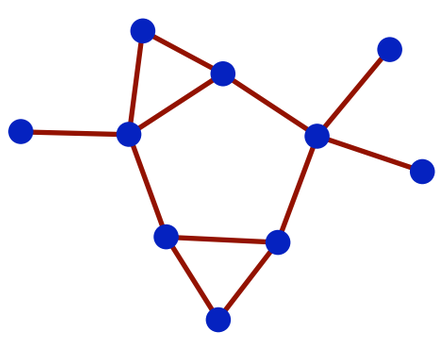
$\mathrm{Aut}(\Gamma_1) \cong S_5$

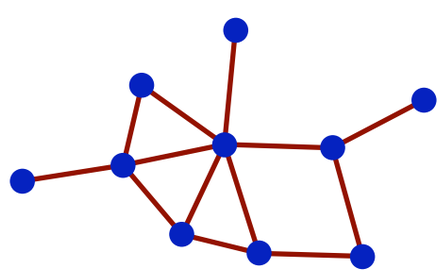
$\mathrm{Aut}(\Gamma_2) \cong \mathbb{Z}_2$

Graph automorphism

Problem: Given an n-vertex graph Γ, what is its automorphism group?

$\Gamma_1 =$ 

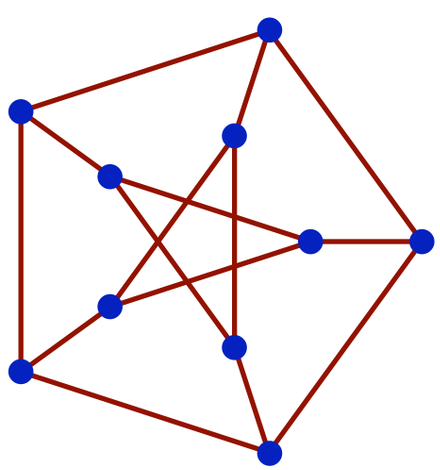
$\Gamma_2 =$ 

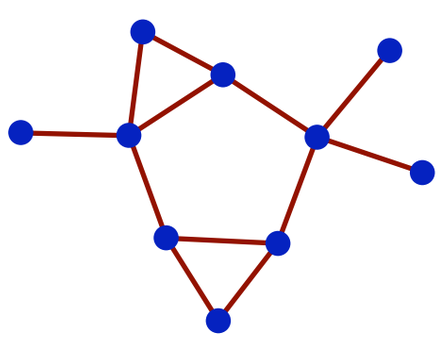
$\Gamma_3 =$ 

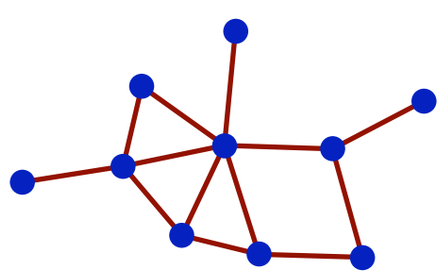
$\mathrm{Aut}(\Gamma_1) \cong S_5$ $\mathrm{Aut}(\Gamma_2) \cong \mathbb{Z}_2$ $\mathrm{Aut}(\Gamma_3) = \{\mathrm{id}\}$

Graph automorphism

Problem: Given an n-vertex graph Γ, what is its automorphism group?

$\Gamma_1 =$ 

$\Gamma_2 =$ 

$\Gamma_3 =$ 

$\mathrm{Aut}(\Gamma_1) \cong S_5$ $\mathrm{Aut}(\Gamma_2) \cong \mathbb{Z}_2$ $\mathrm{Aut}(\Gamma_3) = \{\mathrm{id}\}$

This is an HSP in $G = S_n$

with $f(\pi) = \pi(\Gamma)$

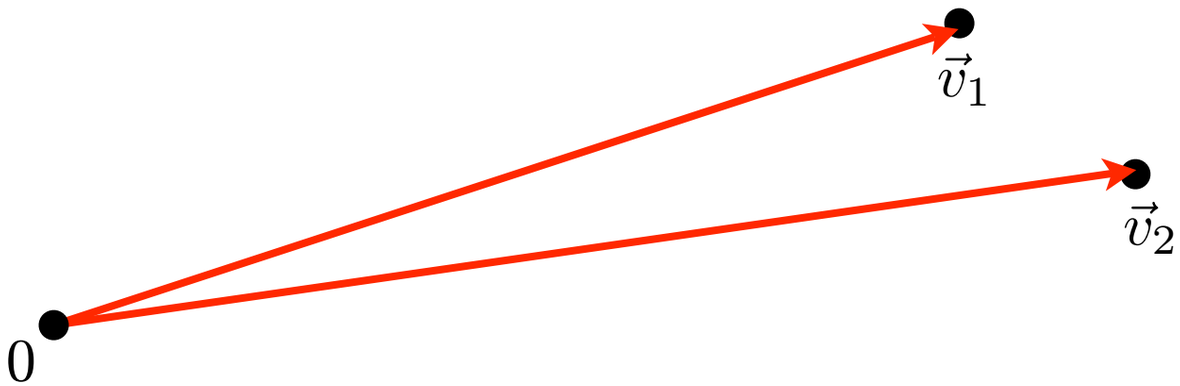
having hidden subgroup $H = \mathrm{Aut}(\Gamma)$

Shortest lattice vector problem

Basis vectors $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ define a lattice $L := \{\sum_i c_i \vec{v}_i : c_i \in \mathbb{Z}\}$

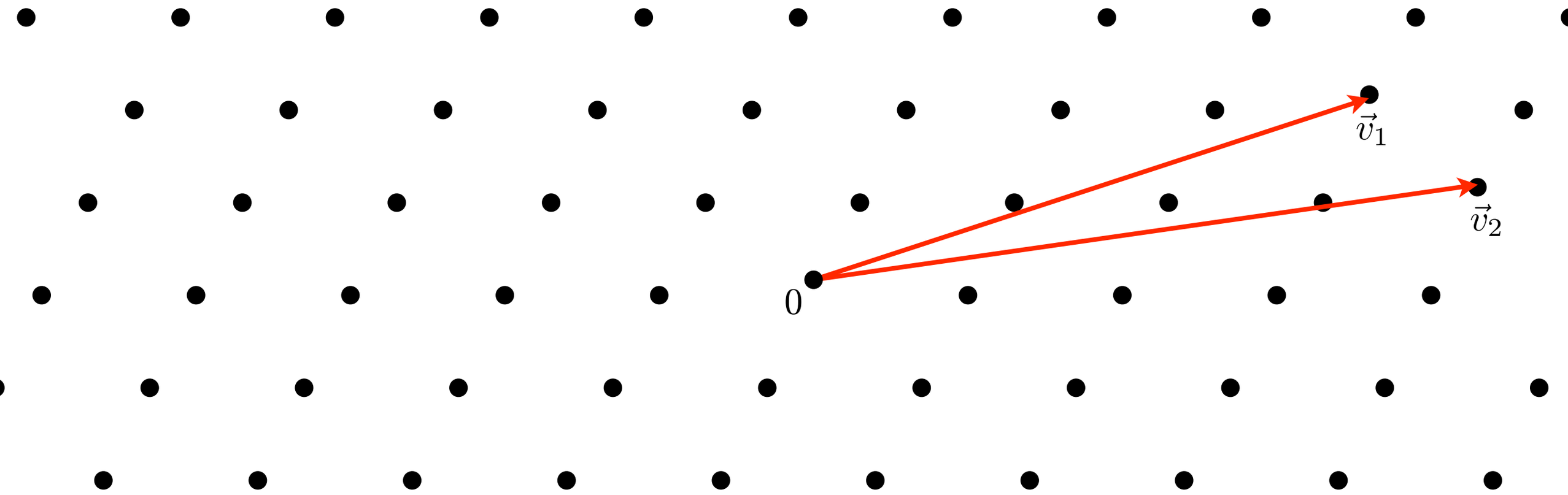
Shortest lattice vector problem

Basis vectors $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ define a lattice $L := \{\sum_i c_i \vec{v}_i : c_i \in \mathbb{Z}\}$



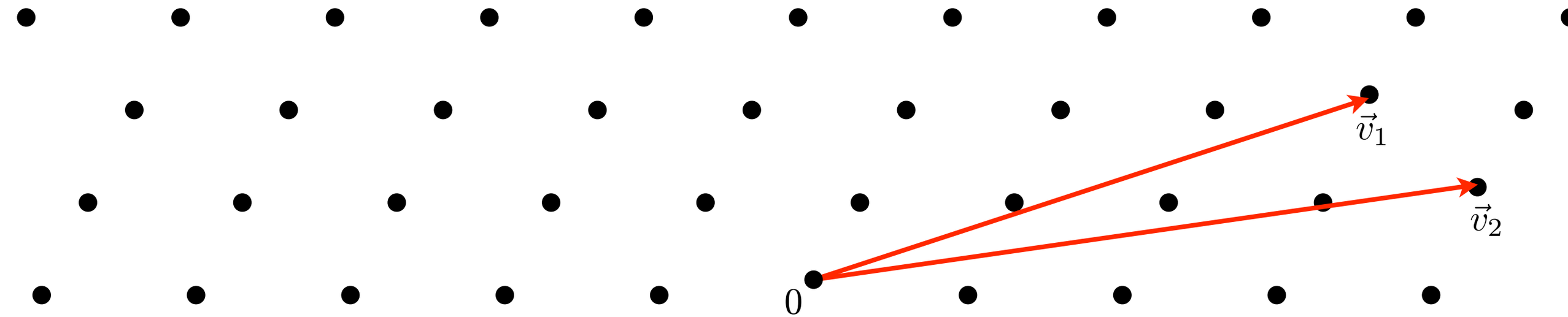
Shortest lattice vector problem

Basis vectors $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ define a lattice $L := \{\sum_i c_i \vec{v}_i : c_i \in \mathbb{Z}\}$



Shortest lattice vector problem

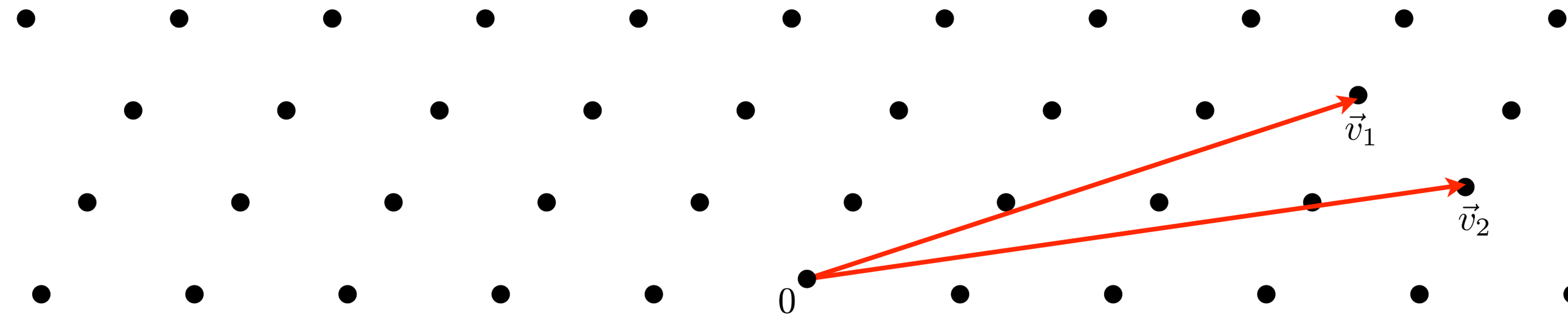
Basis vectors $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ define a lattice $L := \{\sum_i c_i \vec{v}_i : c_i \in \mathbb{Z}\}$



Problem: What is the shortest nonzero vector in the lattice?

Shortest lattice vector problem

Basis vectors $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ define a lattice $L := \{\sum_i c_i \vec{v}_i : c_i \in \mathbb{Z}\}$

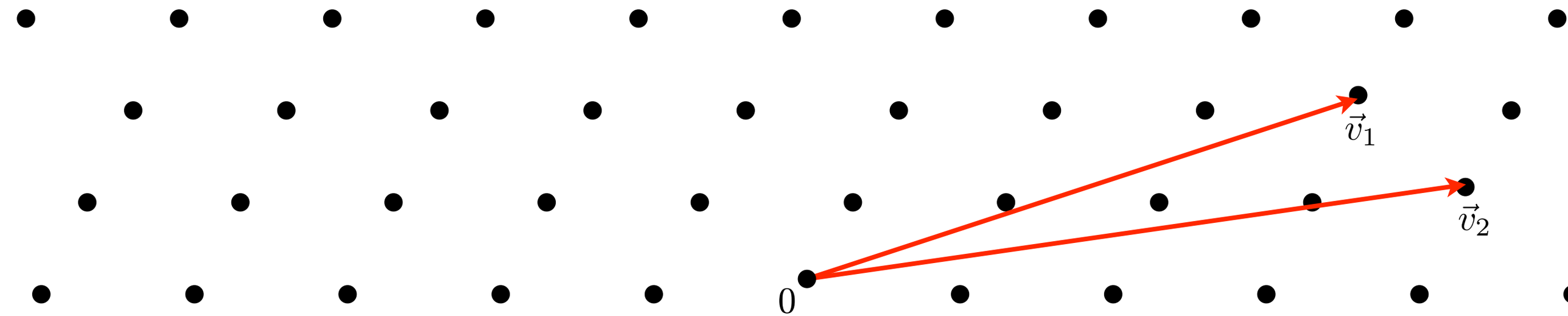


Problem: What is the shortest nonzero vector in the lattice?

This is NP-hard, but what if we promise that the shortest vector is shorter than the next-shortest non-parallel vector by some factor?

Shortest lattice vector problem

Basis vectors $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ define a lattice $L := \{\sum_i c_i \vec{v}_i : c_i \in \mathbb{Z}\}$



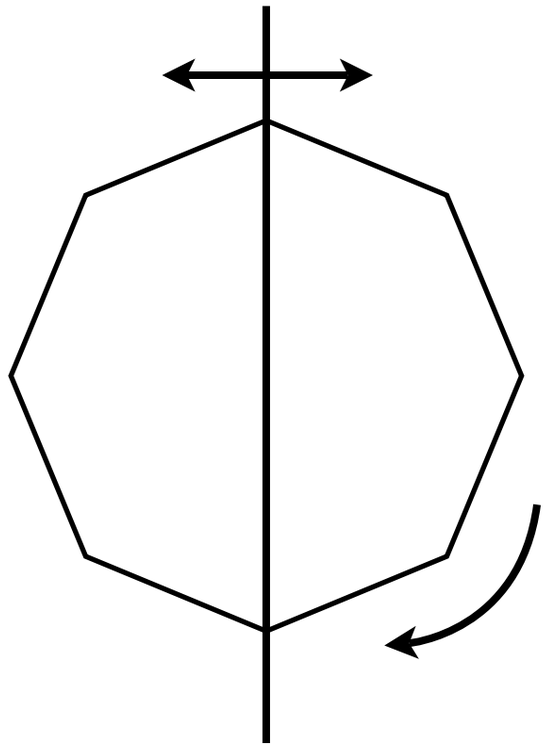
Problem: What is the shortest nonzero vector in the lattice?

This is NP-hard, but what if we promise that the shortest vector is shorter than the next-shortest non-parallel vector by some factor?

Lattice cryptosystems assume this is hard with a factor $\mathrm{poly}(n)$.

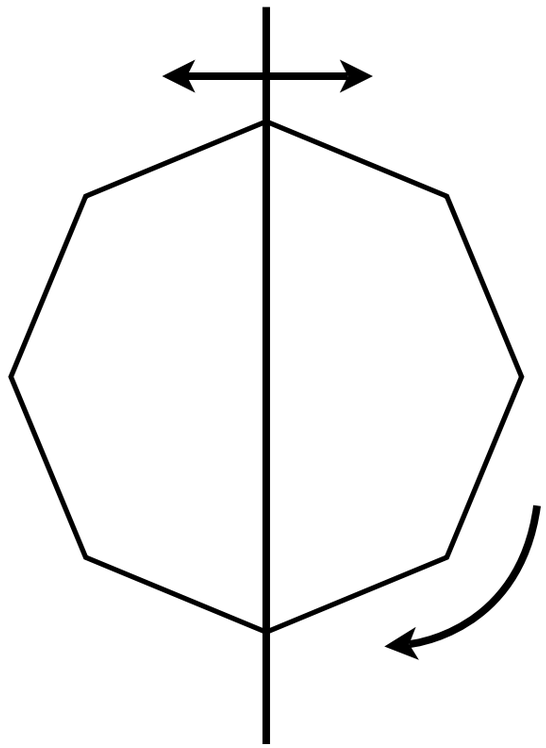
Dihedral hidden subgroup problem

The *dihedral group* of order $2N$ is the group of symmetries of an N-sided regular polygon.



Dihedral hidden subgroup problem

The *dihedral group* of order $2N$ is the group of symmetries of an N-sided regular polygon.

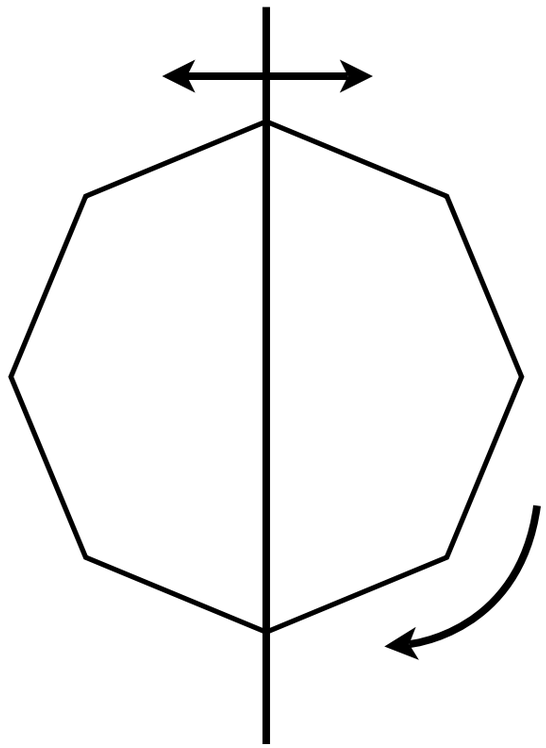


Is there an efficient quantum algorithm for the HSP in this group?

Note: It suffices to consider hidden subgroups of order 2 (hidden reflections).

Dihedral hidden subgroup problem

The *dihedral group* of order $2N$ is the group of symmetries of an N-sided regular polygon.



Is there an efficient quantum algorithm for the HSP in this group?

Note: It suffices to consider hidden subgroups of order 2 (hidden reflections).

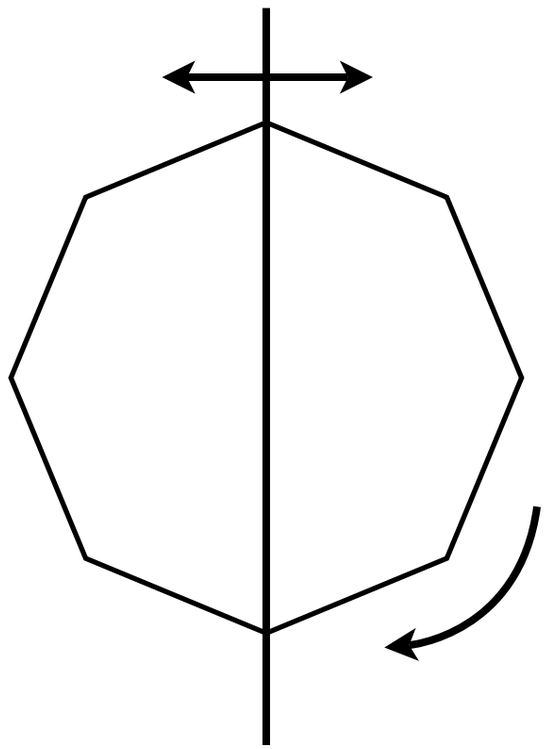
"Standard method":

Given samples of $|\phi_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i s x/N}|1\rangle)$

with x uniformly random in $\mathbb{Z}_N$ (and known), determine s.

Dihedral hidden subgroup problem

The *dihedral group* of order $2N$ is the group of symmetries of an N-sided regular polygon.



Is there an efficient quantum algorithm for the HSP in this group?

Note: It suffices to consider hidden subgroups of order 2 (hidden reflections).

"Standard method":

Given samples of $|\phi_x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i s x/N}|1\rangle)$
with x uniformly random in $\mathbb{Z}_N$ (and known), determine s.

Solving this problem would break lattice-based cryptography, one of the few types of cryptosystems not yet known to be broken by quantum computers!