Finding Patterns

① Make superposition of all inputs

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}|x\rangle$$

Finding Patterns

① Make superposition of all inputs

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}|x\rangle$$

② Get answers in the amplitude

B_f gives $\frac{1}{\sqrt{N}}\sum_x (-1)^{f(x)}|x\rangle$

Call $F(x) = (-1)^{f(x)}$

$F: \{0,1\}^n \longrightarrow \{\pm 1\}$

$0 \longrightarrow 1$

$1 \longrightarrow -1$

Loading up data in the vector

$$\frac{1}{\sqrt{N}}\begin{bmatrix} F(00\ldots0) \\ F(00\ldots1) \\ \vdots \\ F(11\ldots1) \end{bmatrix}$$

Finding Patterns

① Make superposition of all inputs

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}|x\rangle$$

② Get answers in the amplitude

B_f gives $\frac{1}{\sqrt{N}}\sum_x (-1)^{f(x)}|x\rangle$

Call $F(x) = (-1)^{f(x)}$

$F: \{0,1\}^n \longrightarrow \{\pm 1\}$

$0 \longrightarrow 1$

$1 \longrightarrow -1$

③ Create interference

$H^{\otimes n}$ again

$$H^{\otimes n}\left(\frac{1}{\sqrt{N}}\sum_x F(x)|x\rangle\right)$$

$$= \frac{1}{\sqrt{N}}\sum_x F(x)\, H^{\otimes n}|x\rangle = \frac{1}{\sqrt{N}}\sum_s \;?\; |s\rangle$$

Loading up data in the vector

$$\frac{1}{\sqrt{N}}\begin{bmatrix} F(00..0) \\ F(00...1) \\ \vdots \\ F(11...1) \end{bmatrix}$$

Finding Patterns

① Make superposition of all inputs ② Get answers in the amplitude

③ Create interference

The Boolean Fourier Transform

$H^{\otimes n}$ does the job for us

if the pattern we are looking for is of an XOR function

Data vector of length N → Can be any orthonormal basis for $\mathbb{R}^N$ $\{|x_0\rangle, |x_1\rangle, \dots, |x_{N-1}\rangle\}$ → s^{th} entry of length N vector identifies "strength" of s^{th} pattern in the data

(think of them as N pattern vectors)

Classically we have a physical vector of size N

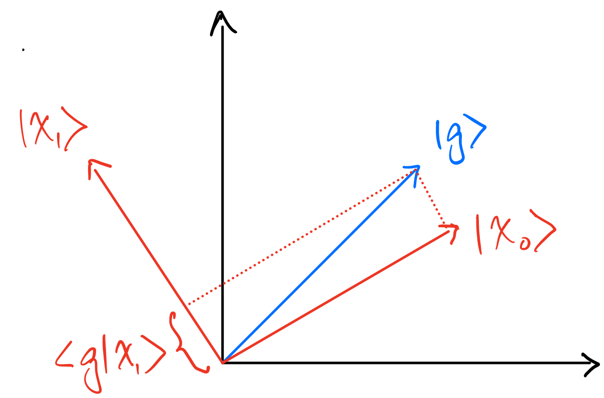
Qtm we benefit by having $N = 2^n$

Def: For any $g : \{0,1\}^n \in \mathbb{R}$, $|g\rangle$ denotes $\frac{1}{\sqrt{N}} \sum_x g(x) |x\rangle$

$|g\rangle$ is a qtm state iff $\frac{1}{N} \sum_x g(x)^2 = 1$

"Strength of pattern" in $|g\rangle$ given by coefficients of $|g\rangle$ when represented in $|\chi_s\rangle$ basis.

"Strength of $|\chi_s\rangle$": $\langle \chi_s | g \rangle$



Decompose $g : \{0,1\}^n \longrightarrow \mathbb{R}$ into basis of XOR functions

$$\chi_s : \{0,1\}^n \longrightarrow \{\pm 1\}$$
$$x \longmapsto (-1)^{s \cdot x}, \quad s \in \{0,1\}^n$$

$s_1 \cdot x_1 \oplus s_2 \cdot x_2$

$$s \cdot x = s_1 x_1 \oplus s_2 x_2 \oplus \dots \oplus s_n x_n$$

$(-1)^{s.x}$

for $n=2$

$$\begin{array}{c|cccc} |x\rangle \backslash |s\rangle & |00\rangle & |01\rangle & |10\rangle & |11\rangle \\ \hline |00\rangle & & & & \\ |01\rangle & & & & \\ |10\rangle & & & & \\ |11\rangle & & & & \end{array} \quad \underbrace{\frac{1}{2}\begin{pmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{pmatrix}}_{H \otimes H}$$

Build χ for $n=1$, $N=2$

$$|x\rangle \begin{cases} |0\rangle \\ |1\rangle \end{cases} \quad \overset{\chi_{s=0}}{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}}, \quad \overset{\chi_{s=1}}{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}}$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Property of XOR pattern function $\chi_s(x) = (-1)^{s \cdot x}$

$$\chi_s(x+y) = \chi_s(x)\chi_s(y)$$

$$(-1)^{s\cdot(x+y)} = (-1)^{s\cdot x + s\cdot y} = (-1)^{s\cdot x}(-1)^{s\cdot y} = \chi_s(x)\chi_s(y)$$

Generalizing for $x, s \in \mathbb{Z}_N = \{0, 1, \dots, N-1\}$ $\qquad$ $\chi_s(x+y) = \chi_s(x)\chi_s(y)$

i) $\chi_s(x+0) = \chi_s(x)\chi_s(0)$

$\Rightarrow \chi_s(0) = 1$ for all s

ii) $\chi_s(\underbrace{x + x + \dots + x}_{N \text{ times}}) = \chi_s(x)\,\chi_s(x) \dots \chi_s(x) = \chi_s(x)^N$

$$\Rightarrow \chi_s(Nx \bmod N) = \chi_s(0) = \underbrace{1 = \chi_s(x)^N}_{N^{th} \text{ roots of unity}}$$

$$\chi_s(x) = e^{2\pi i s x/N}$$

Simon's Algorithm

$$f : \{0,1\}^n \to \{0,1\}^n$$

f is promised to have the property:

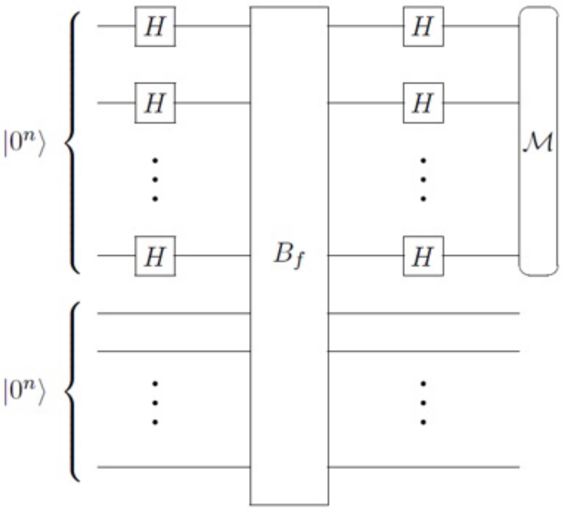
$$[f(x) = f(y)] \Leftrightarrow [x \oplus y \in \{0^n, s\}]$$

$x \oplus y = s$

$x = s \oplus y$

i.e., there exists a string s, such that

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



Classical Complexity $\Omega\left(\sqrt{2^n}\right)$

Simon's Algorithm

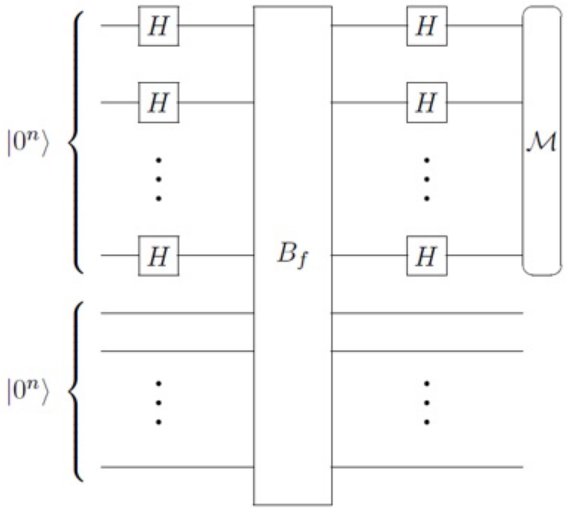
$$f : \{0,1\}^n \to \{0,1\}^n$$

f is promised to have the property:

$$[f(x) = f(y)] \Leftrightarrow [x \oplus y \in \{0^n, s\}] \quad \to x \oplus y = s$$

i.e., there exists a string s, such that

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



Example, $n = 3$

x	$f(x)$
000	101
001	010
010	000
011	110
100	000
101	110
110	101
111	010

$f(000) = 101 = f(110)$

$s = x \oplus y \to 000 \oplus 110 = 110$

$f(001) = 010$

$x \to x \oplus s = 001 \oplus 110 = 111 = y$

$f(11) = 010$

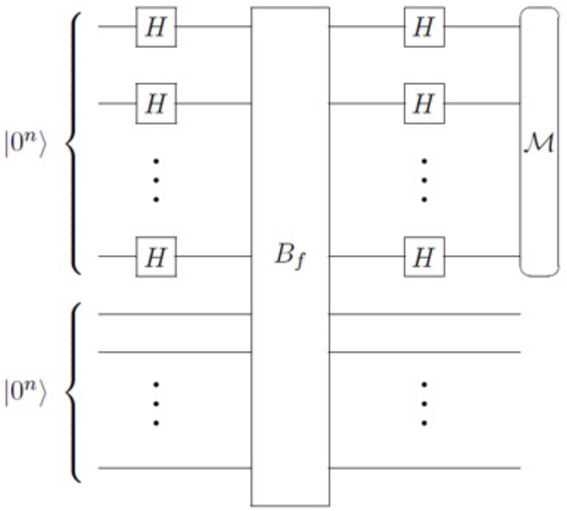
Simon's Algorithm

$$|0^n\rangle|0^n\rangle \xrightarrow{H^{\otimes n}\otimes \mathbb{1}} \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|0^n\rangle$$

$$\xrightarrow{B_f} \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|f(x)\rangle$$

$$\xrightarrow{H^{\otimes n}\otimes \mathbb{1}} \frac{1}{2^n}\sum_x\sum_y (-1)^{x\cdot y}|y\rangle|f(x)\rangle$$

$$\sum_y |y\rangle\left(\frac{1}{2^n}\sum_x (-1)^{x\cdot y}|f(x)\rangle\right)$$



<u>if $s = 0^n$</u>

$$\left\|\frac{1}{2^n}\sum_x (-1)^{x\cdot y}|f(x)\rangle\right\|^2 = \frac{1}{2^n}$$

Simon's Algorithm

$$\left\| \frac{1}{2^n} \sum_x (-1)^{x\cdot y} |f(x)\rangle \right\|^2$$

if $s \neq 0^n$

$f(x_z) = z = f(x'_z)$, $x_z \oplus x'_z = s$, $z, x_z, x'_z \in \{0,1\}^n$, $s \neq 0^n$

Let A be the range of f. If $z \in A$, then there exist two unique strings x_z & x'_z, s.t.

$$\left\| \frac{1}{2^n} \sum_{z\in A} \left((-1)^{x_z\cdot y} + (-1)^{x'_z\cdot y} \right) |z\rangle \right\|^2$$

$$\underbrace{(x_z \oplus x'_z)}_{s} \cdot y$$

$$= \left\| \frac{1}{2^n} \sum_{z\in A} (-1)^{x_z\cdot y} \left(1 + (-1)^{x'_z\cdot y \oplus x_z\cdot y} \right) |z\rangle \right\|^2$$

$$= \left\| \frac{1}{2^n} \sum_{z} (-1)^{x_z\cdot y} \left(1 + (-1)^{s\cdot y} \right) |z\rangle \right\|^2$$

Simon's Algorithm

$$\left\| \frac{1}{2^n} \sum_z (-1)^{x_z \cdot y} \left(1 + (-1)^{s \cdot y}\right) |z\rangle \right\|^2$$

$$= \begin{cases} 1/2^{n-1} & \text{if } s \cdot y = 0 \\ 0 & \text{if } s \cdot y = 1 \end{cases}$$

$$s_1 y_1 + s_2 y_2 + \cdots + s_n y_n$$

Simon's Algorithm

Classical Post-Processing

$y_i \in \{0,1\}^n$

Repeat Simon's circuit $O(n)$ times to obtain strings $y_1, y_2, \ldots, y_n$, all linearly independent.

$$\left.\begin{aligned} s \cdot y_1 &= 0 \\ s \cdot y_2 &= 0 \\ &\vdots \\ s \cdot y_n &= 0 \end{aligned}\right\}$$

System of n equations with n unknowns $(s_1, s_2, \ldots, s_n)$

$s_i \in \{0,1\}$