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

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

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 $\mathbb{R}^N$ $\{|\chi_0\rangle, |\chi_1\rangle, \ldots, |\chi_{N-1}\rangle\}$ ⟶ 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

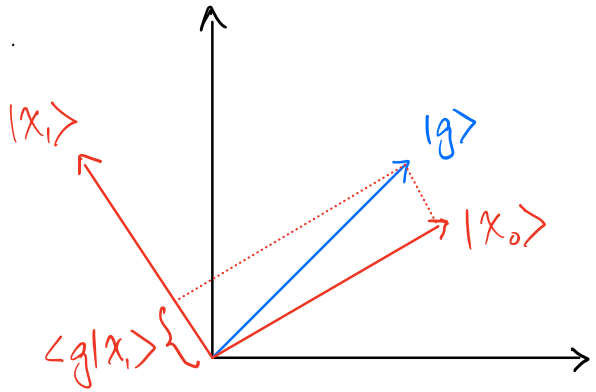
Qtm we benefit by having $N = 2^n$

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 $|x_s\rangle$ basis.

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



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

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

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

Build χ for $n=1$, $N=2$

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

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

$(-1)^{s.x}$ for $n=2$

$\lvert x\rangle$ \ $\lvert s\rangle$	$\lvert 00\rangle$	$\lvert 01\rangle$	$\lvert 10\rangle$	$\lvert 11\rangle$
$\lvert 00\rangle$	$+1$	$+1$	$+1$	$+1$
$\lvert 01\rangle$	$+1$	-1	$+1$	-1
$\lvert 10\rangle$	$+1$	$+1$	-1	-1
$\lvert 11\rangle$	$+1$	-1	-1	$+1$

$\frac{1}{2}$

$H \otimes H$

Simon's Algorithm

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

f is promised to have the property:

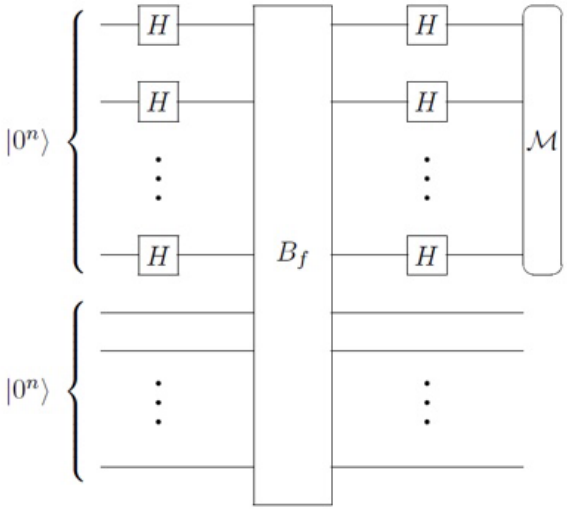
$$[f(x) = f(y)] \Leftrightarrow [x \oplus y \in \{0^n, s\}]$$

$x \oplus y = s$

$x = s \oplus y$

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

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



Simon's Algorithm

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

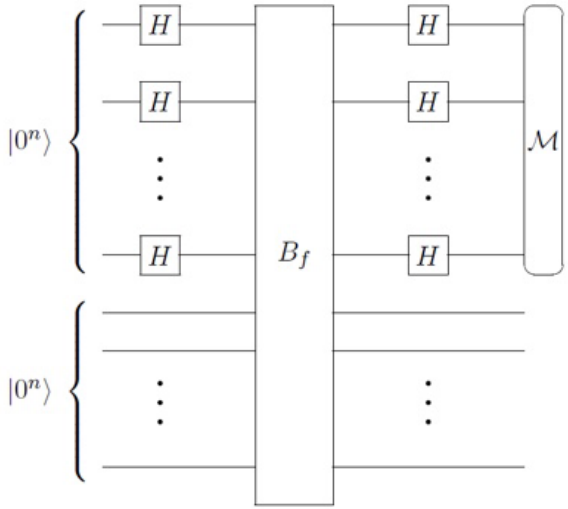
f is promised to have the property:

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

$\rightarrow x \oplus y = s$

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

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

$x = 000$, $y = 110$

$s = x \oplus y \Rightarrow 110$

$x' \oplus s = y'$

$001 \oplus 110 = 111 \rightarrow f(x') = f(y')$

Classical Query Complexity

$\Omega(\sqrt{2^n})$

Simon's Algorithm

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

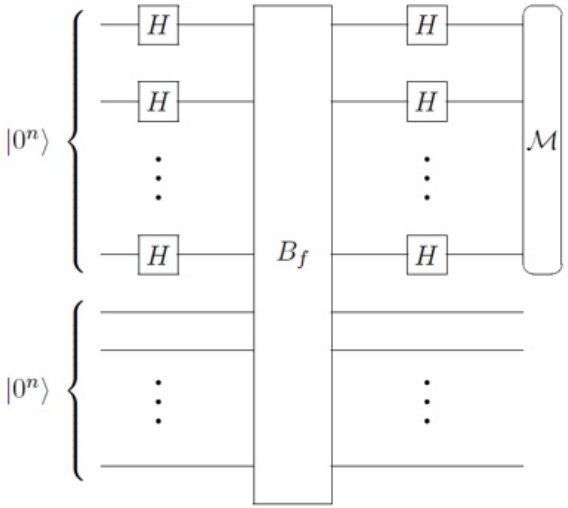
$$\xrightarrow{B_f} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle$$

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

if $s = 0^n$

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

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



Simon's Algorithm

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

if $s \neq 0^n$

Let A be the range of f. If $z \in A$, $z \in \{0,1\}^n$, then there exist two unique strings $x_z, x'_z \in \{0,1\}^n$ s.t.

$$z = f(x_z) = f(x'_z), \qquad x_z \oplus x'_z = s$$

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

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

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

Simon's Algorithm

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

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

$\longrightarrow$ can create a system of equations with different y.

Simon's Algorithm

Classical Post-Processing

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

Repeat the circuit $O(n)$ times, to obtain strings $y_1, y_2, \ldots, y_n$, s.t.

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

The y_i's should be linearly independent.

for any $\varepsilon > 0$, we can solve with error at most ε using $O(n)$ queries.