Simon's Algorithm

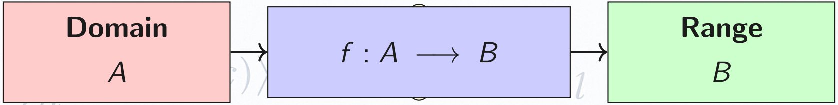
Tilock Sadhukhan

C-DAC Bengaluru

27 August, 2025

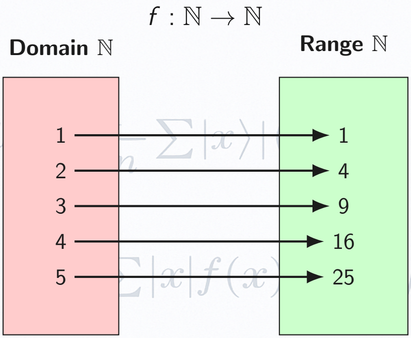
What is a Function?

A rule that assigns to each input to an output.

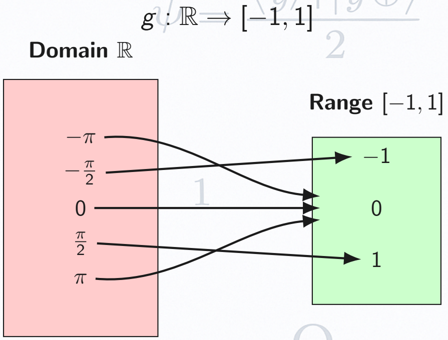


Examples

- $f : \mathbb{N} \to \mathbb{N}, \quad f(x) = x^2$. Domain: natural numbers; Range: perfect squares.
- $g : \mathbb{R} \to [-1, 1], \quad g(x) = \sin(x)$. Domain: real numbers; Range: sine values.

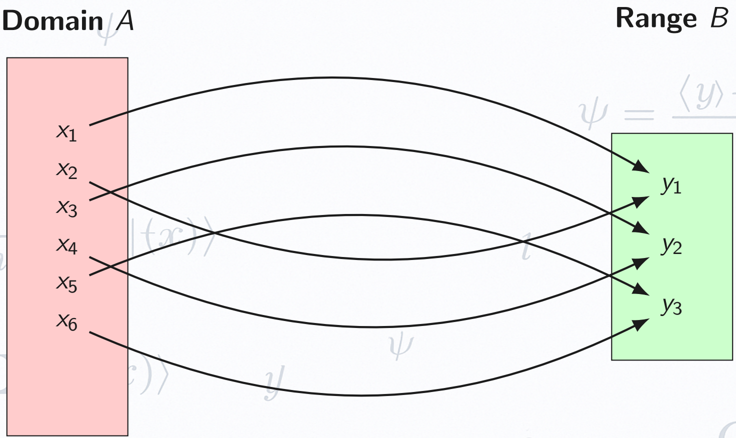


Function: $f(x) = x^2$



Function: $g(x) = \sin x$

Illustration of a 2-to-1 Function



Problem Setup

We are given a 2-to-1 function $f : \{0,1\}^n \to \{0,1\}^n$ for which there exists a secret string $s \in \{0,1\}^n$ such that for all inputs $x \in \{0,1\}^n$: $\quad f(x) = f(x \oplus s)$.

Problem Setup

We are given a 2-to-1 function $f : \{0,1\}^n \to \{0,1\}^n$ for which there exists a secret string $s \in \{0,1\}^n$ such that for all inputs $x \in \{0,1\}^n$: $\quad f(x) = f(x \oplus s)$.

Goal: To find the secret string s using oracle queries to f.

Classical Approach

As a concrete example, let us assume $n = 3$ and $s = 101$. The function's values might be given by,

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

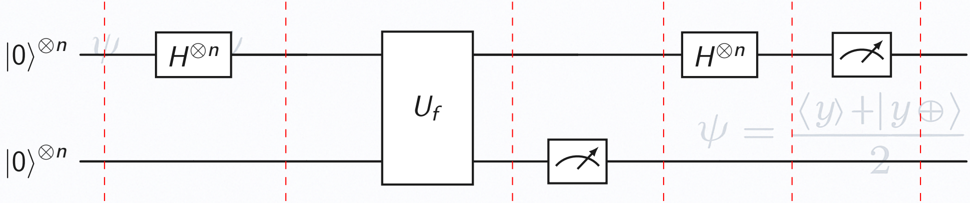
Classical Approach

As a concrete example, let us assume $n = 3$ and $s = 101$. The function's values might be given by,

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

Here, the total number of inputs is $2^n = N$. To solve this problem using a classical computer, we need to input values one by one until we encounter a repeated output. In the worst case, the maximum number of inputs required is half of 2^{n-1} plus one; that is, we may need to check up to $2^{n-1} + 1$ inputs.

Circuit skeleton (two n-qubit registers)



- Start in $|\psi_0\rangle = |0\rangle^{\otimes n}|0\rangle^{\otimes n}$.
- First Hadamard on top: $|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |0\rangle^{\otimes n}$.
- Oracle query U_f: $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle \rightarrow |\psi_2\rangle$.
- Measure bottom $\rightarrow$ collapse to $|\psi_3\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus s\rangle)$.
- Second Hadamard & measure top $\rightarrow$ outcome $|\psi_4\rangle$ satisfying $w \cdot s = 0$.

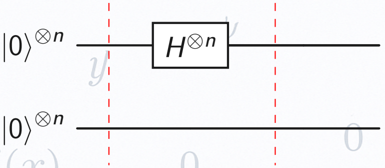
Step 1: Create a uniform superposition

We begin with both registers in $|0\rangle^{\otimes n}$ and the initial state as $|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n}$. By applying a Hadamard gate H to each qubit ($H^{\otimes n}$) on the top gives the uniform superposition $|\psi_1\rangle$.

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle .$$

So,

$$|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n} \xrightarrow{H^{\otimes n} \otimes I} |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |0\rangle^{\otimes n} .$$



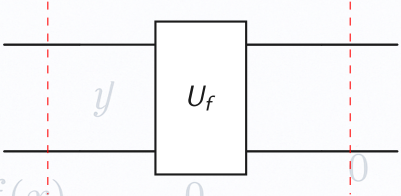
Step 2: Query the oracle U_f

The oracle U_f is a black-box unitary that implements the function f via bitwise XOR:

$$U_f : |x\rangle\,|y\rangle \;\mapsto\; |x\rangle\,|\,y \oplus f(x)\,\rangle\,.$$

Starting from the uniform superposition $|\psi_1\rangle$, this entangles the two registers:

$$|\psi_2\rangle = U_f\;|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \;\otimes\; |f(x)\rangle\,.$$



Step 3: Measure the bottom register

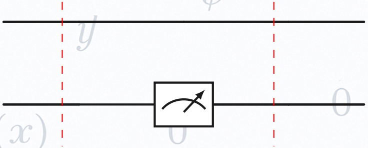
We had,

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes |f(x)\rangle .$$

Now, measuring the bottom register gives some outcome $f(x_0)$. Since $f(x_0) = f(x_0 \oplus s)$, the top register collapses to

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}\big(|x_0\rangle + |x_0 \oplus s\rangle\big).$$

This "hides" the unknown s in the superposition.



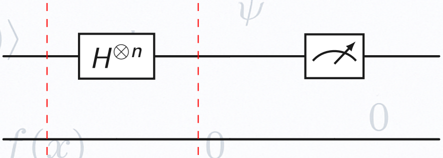
Step 4: Second Hadamard and Readout

We start from the post-measurement state, $|\psi_3\rangle = \frac{1}{\sqrt{2}}\big(|x_0\rangle + |x_0 \oplus s\rangle\big)$. Applying $H^{\otimes n}$ to this gives

$$|\psi_4\rangle = (H^{\otimes n} \otimes I)\,|\psi_3\rangle = \frac{1}{\sqrt{2^{\,n+1}}} \sum_{w\in\{0,1\}^n} \Big[(-1)^{x_0\cdot w} \; + \; (-1)^{(x_0\oplus s)\cdot w}\Big]\,|w\rangle\,.$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{w} (-1)^{x_0\cdot w}\left[1 + (-1)^{s\cdot w}\right]|w\rangle\,.$$

Because $(x_0 \oplus s)\cdot w = x_0 \cdot w \oplus s \cdot w$, the two terms cancel unless $s \cdot w = 0$. Measuring the top register therefore yields a random w satisfying $w \cdot s = 0$.



Step 5: From amplitudes to the constraint $w \cdot s = 0$

We know

$$(x \oplus y) \cdot w = x \cdot w \oplus y \cdot w \tag{1}$$

$$(-1)^{u \oplus v} = (-1)^{u} (-1)^{v} \tag{2}$$

If we start from

$$\sum_{w} \left[(-1)^{x_0 \cdot w} + (-1)^{(x_0 \oplus s) \cdot w} \right] |w\rangle \, .$$

Using (1) and (2), we get,

$$(-1)^{(x_0 \oplus s) \cdot w} = (-1)^{x_0 \cdot w \,\oplus\, s \cdot w} = (-1)^{x_0 \cdot w} (-1)^{s \cdot w} .$$

Now factoring out $(-1)^{x_0 \cdot w}$:

$$(-1)^{x_0 \cdot w} + (-1)^{(x_0 \oplus s) \cdot w} = (-1)^{x_0 \cdot w} \left[1 + (-1)^{s \cdot w} \right] .$$

Therefore

$$|\psi_4\rangle = \frac{1}{\sqrt{2}^{\,n+1}} \sum_{w} (-1)^{x_0 \cdot w} \left[1 + (-1)^{s \cdot w} \right] |w\rangle \, .$$

Outcomes from $|\psi_4\rangle$: cases $s \cdot w = 0$ vs 1

Start from

$$|\psi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_w (-1)^{x_0 \cdot w} \left[1 + (-1)^{s \cdot w}\right] |w\rangle .$$

Let the amplitude of basis state $|w\rangle$ be

$$A(w) = \frac{1}{\sqrt{2^{n+1}}} (-1)^{x_0 \cdot w} [1 + (-1)^{s \cdot w}] .$$

Evaluate the bracket:

$$1 + (-1)^{s \cdot w} = \begin{cases} 2, & s \cdot w = 0 \pmod 2, \\ 0, & s \cdot w = 1 \pmod 2. \end{cases}$$

Hence

$$A(w) = \begin{cases} \pm \dfrac{1}{\sqrt{2^{n-1}}}, & s \cdot w = 0, \\ 0, & s \cdot w = 1, \end{cases} \qquad P(w) = |A(w)|^2 = \begin{cases} 2^{-(n-1)}, & s \cdot w = 0, \\ 0, & s \cdot w = 1. \end{cases}$$

Step 6: Recovering the secret string s

What the measurements give. Each run returns a bit string $w \in \{0,1\}^n$ with

$$w \cdot s = \sum_{i=1}^{n} w_i s_i \equiv 0 \pmod 2.$$

How to get s.

- Repeat the experiment until you have about n *independent* strings $w^{(1)}, \dots, w^{(m)}$.
- Stack them as rows of a matrix $W \in \{0,1\}^{m \times n}$.
- Solve the homogeneous system $W\,s = 0 \pmod 2$ (same as ordinary Gaussian elimination, but addition is XOR).
- The nonzero solution of this system is the hidden string s. If the solution is not unique, collect another w and solve again.

Result. s is the unique nonzero vector orthogonal to all observed w's over $\mathbb{F}_2$.

Gaussian Elimination

System of equations	Row operations	Augmented matrix
$\begin{aligned} 2x + y - z &= 8 \\ -3x - y + 2z &= -11 \\ -2x + y + 2z &= -3 \end{aligned}$	(start)	$\left[\begin{array}{ccc\|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right]$
$\begin{aligned} 2x + y - z &= 8 \\ \tfrac{1}{2}y + \tfrac{1}{2}z &= 1 \\ 2y + z &= 5 \end{aligned}$	$L_2 + \frac{3}{2}L_1 \to L_2, \quad L_3 + L_1 \to L_3$	$\left[\begin{array}{ccc\|c} 2 & 1 & -1 & 8 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 2 & 1 & 5 \end{array}\right]$
$\begin{aligned} 2x + y - z &= 8 \\ \tfrac{1}{2}y + \tfrac{1}{2}z &= 1 \\ -z &= 1 \end{aligned}$	$L_3 - 4L_2 \to L_3$	$\left[\begin{array}{ccc\|c} 2 & 1 & -1 & 8 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 1 \end{array}\right]$

Echelon (upper triangular) form reached

Gaussian Elimination

System of equations	Row operations	Augmented matrix
$2x + y = 7$ $\frac{1}{2}y = \frac{3}{2}$ $-z = 1$	$L_1 - L_3 \to L_1, \quad L_2 + \frac{1}{2}L_3 \to L_2$	$\left[\begin{array}{ccc\|c} 2 & 1 & 0 & 7 \\ 0 & \frac{1}{2} & 0 & \frac{3}{2} \\ 0 & 0 & -1 & 1 \end{array}\right]$
$2x + y = 7$ $y = 3$ $z = -1$	$2L_2 \to L_2, \quad -L_3 \to L_3$	$\left[\begin{array}{ccc\|c} 2 & 1 & 0 & 7 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array}\right]$
$x = 2$ $y = 3$ $z = -1$	$L_1 - L_2 \to L_1, \quad \frac{1}{2}L_1 \to L_1$	$\left[\begin{array}{ccc\|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -1 \end{array}\right]$

Solution: $\boxed{x = 2, \; y = 3, \; z = -1}$.

Solving $Ws = 0 \pmod 2$

Goal. Find a binary vector $s = (s_1, s_2, s_3)^\top$ such that $Ws = 0$ over $\mathbb{F}_2$ (all arithmetic is XOR).

$$W = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \qquad s = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}.$$

Idea. Use Gaussian elimination *mod 2* to reduce W and read off constraints on s.

XOR rules & allowed row operations (mod 2)

XOR rules: $0 \oplus 0 = 0,\ 1 \oplus 0 = 1,\ 1 \oplus 1 = 0.$

Allowed row operations (all mod 2):

- Swap rows: $R_i \leftrightarrow R_j$.
- Row addition (XOR): $R_i \leftarrow R_i \oplus R_j$.
- (No scaling—1 is the only nonzero scalar in $\mathbb{F}_2$.)

Goal of elimination: Make leading 1's (pivots) go down/right and clear their columns using XOR.

Row reduction (mod 2)

Start with

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Step 1 (make a pivot in col 1): swap $R_1 \leftrightarrow R_2$

$$\Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Step 2 (clear col 1 below the pivot): $R_3 \leftarrow R_3 \oplus R_1$

$$[1,0,1] \oplus [1,1,1] = [0,1,0] \Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Row reduction (mod 2)

Current matrix:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Step 3 (clear col 2 below the pivot in Row 2): $R_3 \leftarrow R_3 \oplus R_2$

$$[0,1,0] \oplus [0,1,0] = [0,0,0] \Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Step 4 (clean col 2 above the pivot): $R_1 \leftarrow R_1 \oplus R_2$

$$[1,1,1] \oplus [0,1,0] = [1,0,1] \Rightarrow \boxed{\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}$$

This is row-echelon (essentially RREF) over $\mathbb{F}_2$.

Read the equations and solve for s

From the reduced rows:

$$\text{Row 1: } s_1 \oplus s_3 = 0 \;\Rightarrow\; s_1 = s_3, \qquad \text{Row 2: } s_2 = 0.$$

Free variable: column 3 (no pivot) $\Rightarrow$ let $s_3 = t \in \{0, 1\}$.

$$\Rightarrow\; s = (s_1, s_2, s_3) = (t,\, 0,\, t).$$

Nonzero solution (Simon): choose $t = 1 \Rightarrow \boxed{s = 101}$.

Quick verification: does $Ws = 0$?

$$W = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \quad s = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \Rightarrow Ws = \begin{bmatrix} 0 \cdot 1 \oplus 1 \cdot 0 \oplus 0 \cdot 1 \\ 1 \cdot 1 \oplus 1 \cdot 0 \oplus 1 \cdot 1 \\ 1 \cdot 1 \oplus 0 \cdot 0 \oplus 1 \cdot 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \oplus 0 \oplus 1 \\ 1 \oplus 0 \oplus 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Conclusion: $s = 101$ satisfies $Ws = 0 \pmod 2$.

Worked example ($n = 3$, secret $s = 101$) — Setup

Registers: top (input) $= |0\rangle^{\otimes 3}$, bottom (work) $= |0\rangle^{\otimes 3}$.

Promise about the oracle f: for the hidden string $s = 101$,

$$f(x) = f(x \oplus s) \quad \text{for all } x \in \{0,1\}^3.$$

Step 1: Put the input in a uniform superposition

Start in

$$|\psi_0\rangle = |0\rangle^{\otimes 3} |0\rangle^{\otimes 3}.$$

Apply $H^{\otimes 3}$ to the top register:

$$H^{\otimes 3} |0\rangle^{\otimes 3} = \frac{1}{\sqrt{8}} \sum_{x \in \{0,1\}^3} |x\rangle.$$

So the joint state becomes

$$|\psi_1\rangle = \frac{1}{\sqrt{8}} \sum_{x \in \{0,1\}^3} |x\rangle |000\rangle.$$

Each of the 8 three-bit strings $|x\rangle$ is now equally "present" on the top; the bottom is untouched so far.

Step 2: Query the oracle U_f

Oracle action (XOR form):

$$U_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle .$$

On $|\psi_1\rangle$ this yields

$$|\psi_2\rangle = \frac{1}{\sqrt{8}} \sum_x |x\rangle |f(x)\rangle .$$

Step 3: Measure the bottom register

Measure the bottom and suppose the outcome is $f(010)$. Then the top must be one of the two inputs that map to that value:

$$010 \quad \text{or} \quad 010 \oplus 101 = 111.$$

So the state collapses to

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}\big(|010\rangle + |111\rangle\big).$$

Measuring the output "selects a pair" upstairs. We don't know which member of the pair, so we're left with an equal superposition of the two.

Step 4: Second Hadamards — where the interference happens

Apply $H^{\otimes 3}$ to the top register of $|\psi_3\rangle$. Recall the Walsh–Hadamard identity:

$$H^{\otimes 3}|x\rangle = \frac{1}{\sqrt{8}} \sum_{w \in \{0,1\}^3} (-1)^{x \cdot w} |w\rangle .$$

Hence the amplitude of $|w\rangle$ after the second Hadamards is

$$A(w) \;\propto\; (-1)^{010 \cdot w} + (-1)^{(010 \oplus s) \cdot w}.$$

$$\propto [1 + (-1)^{s \cdot w}]$$

Step 5: Allowed outcomes and their probabilities

Constraint $w \cdot s = 0$ with $s = 101$ means $w_1 = w_3$. The allowed w are:

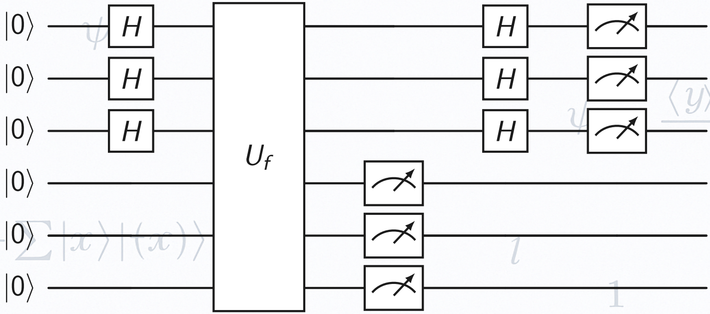
$$\{000,\ 010,\ 101,\ 111\}.$$

Because amplitudes have the same magnitude for all allowed w, the distribution is *uniform* over this 4-element set:

$$P(w) = \begin{cases} \frac{1}{4}, & w \in \{000, 010, 101, 111\}, \\ 0, & \text{otherwise.} \end{cases}$$

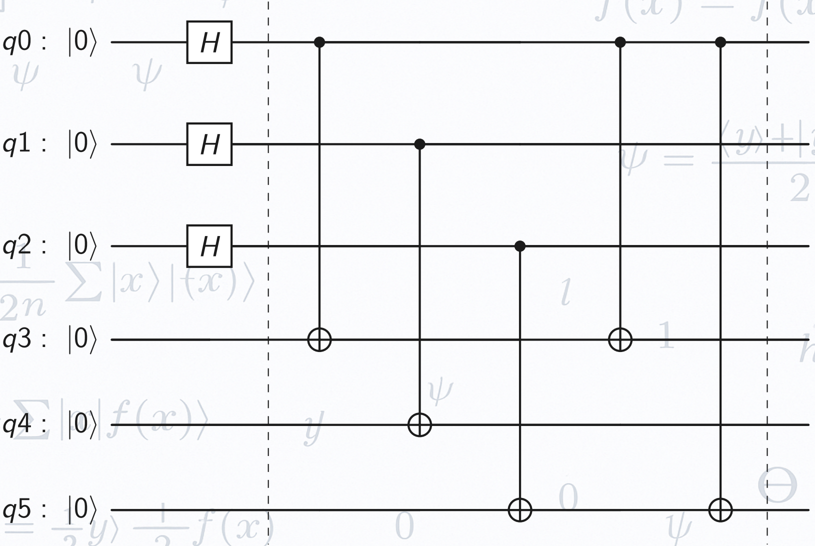
Takeaway: one run gives a random w satisfying $w \cdot s = 0$. Repeat a few times to collect independent equations and solve for s over $\mathbb{F}_2$ and solve the secret string 's' using Gaussian elimination.

Simon's Algorithm for $n = 3$



Quantum circuit for Simon's algorithm with $n = 3$: the top three qubits (input register) are initialized in $|0\rangle$, placed in superposition with Hadamards, processed through the oracle U_f, and then measured after a second Hadamard layer. The bottom three qubits (work register) store the oracle output and are measured but not further transformed.

Oracle wiring (example for $n = 3$)



Oracle action(explicit states)

Order: $|x_0x_1x_2\rangle\,|y_1y_2y_3\rangle$, initial $|y\rangle = |000\rangle$. CNOTs: $x_0 \to y_1$, $x_1 \to y_2$, $x_2 \to y_3$, $x_0 \to y_1$, $x_0 \to y_3$.

$$\psi_{\text{in}} = \tfrac{1}{\sqrt{8}}\big(|000\rangle|000\rangle + |001\rangle|000\rangle + |010\rangle|000\rangle + |011\rangle|000\rangle \\ + |100\rangle|000\rangle + |101\rangle|000\rangle + |110\rangle|000\rangle + |111\rangle|000\rangle\big)$$

$$\psi^{(1)}\ (x_0 \to y_1) = \tfrac{1}{\sqrt{8}}\big(|000\rangle|000\rangle + |001\rangle|000\rangle + |010\rangle|000\rangle + |011\rangle|000\rangle \\ + |100\rangle|100\rangle + |101\rangle|100\rangle + |110\rangle|100\rangle + |111\rangle|100\rangle\big)$$

$$\psi^{(2)}\ (x_1 \to y_2) = \tfrac{1}{\sqrt{8}}\big(|000\rangle|000\rangle + |001\rangle|000\rangle + |010\rangle|010\rangle + |011\rangle|010\rangle \\ + |100\rangle|100\rangle + |101\rangle|100\rangle + |110\rangle|110\rangle + |111\rangle|110\rangle\big)$$

Oracle action(explicit states)

$$\psi^{(3)}\ (x_2 \to y_3) = \frac{1}{\sqrt{8}}\big(|000\rangle|000\rangle + |001\rangle|001\rangle + |010\rangle|010\rangle + |011\rangle|011\rangle$$
$$+ |100\rangle|100\rangle + |101\rangle|101\rangle + |110\rangle|110\rangle + |111\rangle|111\rangle\big)$$

$$\psi^{(4)}\ (x_0 \to y_1) = \frac{1}{\sqrt{8}}\big(|000\rangle|000\rangle + |001\rangle|001\rangle + |010\rangle|010\rangle + |011\rangle|011\rangle$$
$$+ |100\rangle|000\rangle + |101\rangle|001\rangle + |110\rangle|010\rangle + |111\rangle|011\rangle\big)$$

$$\boxed{\begin{aligned}\psi^{(5)}\ (x_0 \to y_3) = \frac{1}{\sqrt{8}}\big(&|000\rangle|000\rangle + |001\rangle|001\rangle + |010\rangle|010\rangle + |011\rangle|011\rangle\\ &+ |100\rangle|001\rangle + |101\rangle|000\rangle + |110\rangle|011\rangle + |111\rangle|010\rangle\big)\end{aligned}}$$

With secret string $s = 101$.

Simon's Algorithm — Summary

Goal

Find the hidden bit string $s \neq 0^n$ such that $f(x) = f(x \oplus s)$ for all x.

Setup: $f : \{0,1\}^n \to \{0,1\}^n$ is 2-to-1 with the promise above.
One run of the circuit:

1. Prepare $|0\rangle^{\otimes n} |0\rangle^{\otimes n}$; apply $H^{\otimes n}$ to the top register.
2. Query the oracle $U_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$.
3. Apply $H^{\otimes n}$ to the top register and measure it to get $w \in \{0,1\}^n$.

Key rule (from interference): Only outcomes with $w \cdot s = 0$ can appear, and they are *uniform* over that subspace. Outcomes with $w \cdot s = 1$ never occur.
Recovering s: Repeat until you have about n independent w's. Stack them as rows of W and solve $Ws = 0$ over bits (row operations are XOR). The unique nonzero solution is s. If a new w is dependent or all-zero, just run again.
Advantage: $O(n)$ quantum queries vs. $\Theta(2^{n/2})$ classical queries.

Practical use cases

Simon's algorithm itself isn't directly used in industrial problems.

Where you'll see it in practice

- Courses, labs, and demos to explain "Quantum Advantage."
- Like Deutsch–Jozsa and Bernstein–Vazirani, Simon's algorithm has no direct practical use but is important as a toy model for understanding advanced quantum algorithms such as Shor's
- It laid the groundwork for Shor's algorithm, which built upon Simon's ideas (Fourier sampling, hidden subgroup problem).
- Research as a toy model for hidden-structure problems and query complexity.

References & Contact

- D. R. Simon, "On the Power of Quantum Computation," *SIAM Journal on Computing* 26(5):1474–1483, 1997.
- M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press, 10th Anniversary Ed.
- Lecture notes: MIT 6.845 / Berkeley CS294 (Simon's problem and HSP).

Contact
tilocks@cdac.in
tilocksadhukhan0123456789@gmail.com