# Simon's Algorithm

Emad Rezaei Fard Boosari

August 28, 2024

## 1 Simon's Problem

In this lecture, we will delve into one of the cornerstone algorithms of quantum computing—Simon's Algorithm. Developed by Daniel Simon in 1994, this algorithm was one of the first to demonstrate the potential of quantum computers to solve specific problems exponentially faster than classical computers. The significance of Simon's Algorithm lies not only in its computational speedup but also in its role in inspiring the development of Shor's Algorithm, which has profound implications in cryptography.

### 1.1 Problem Statement: [1]

**Algorithm 1** Determine String $s$

```
1: Input: A black-box for computing an unknown function $f : \{0,1\}^n \to \{0,1\}^n$.
2: Promise: There exists a string $p = p_1 p_2 \dots p_n$ such that $f(x) = f(y)$ if $x = y$ or $x = y \oplus p$.
3: Problem: Determine the string $p$ by making queries to $f$.
```

We describe the oracle function $f : \{0,1\}^n \to \{0,1\}^n$ with the following properties:

1. $f$ is 2-to-1; specifically, for each $x_1$, there exists exactly one $x_2 = x_1 \oplus p$ such that $f(x_1) = f(x_2)$.
2. $f$ is periodic with period $p$; i.e., $f(x \oplus p) = f(x)$ for all $x \in \{0,1\}^n$.

**Further Remarks:**
In this block we want to show an example of $f$ which is 2 to 1. Without loss of generality, let me consider $p = 110$, and then we can provide the following table

| $x$ | $x \oplus p = y$ | $f(x) = f(y)$ |
|---|---|---|
| 000 | $000 \oplus 110 = 110$ | $f(000) = f(110)$ |
| 001 | $001 \oplus 110 = 111$ | $f(001) = f(111)$ |
| 010 | $010 \oplus 110 = 100$ | $f(010) = f(100)$ |
| 011 | $011 \oplus 110 = 101$ | $f(011) = f(101)$ |
| 100 | $100 \oplus 110 = 010$ | $f(100) = f(010)$ |
| 101 | $101 \oplus 110 = 011$ | $f(101) = f(011)$ |
| 110 | $110 \oplus 110 = 000$ | $f(110) = f(000)$ |
| 111 | $111 \oplus 110 = 001$ | $f(111) = f(001)$ |

This establishes that $f$ is indeed 2-to-1, confirming that for every $x$, there exists exactly one other value $y$

such that $f(x) = f(y)$.

$$f(000) = \begin{cases} x = 000 \\ x = 110 \end{cases} \quad f(001) = \begin{cases} x = 001 \\ x = 111 \end{cases} \quad f(010) = \begin{cases} x = 010 \\ x = 100 \end{cases} \quad f(011) = \begin{cases} x = 011 \\ x = 101 \end{cases} \tag{1}$$

The value of $f(000), f(001)$, $f(010)$, and $f(011)$ can be an arbitrary string.

Suppose we want to find the period $p$, given an unknown oracle $f$.

## 1.2 Classical Solution

Classically, identifying $p$ requires examining $2^n$ potential combinations. Let me consider the example in Further Remarks in previous section to clarify the number of classical queries to find $p$ (unknown binary string). If I evaluate the function $f$ in order I will get

$$f(000) = a, \quad f(001) = b, \quad f(010) = c, \quad f(011) = d \tag{2}$$

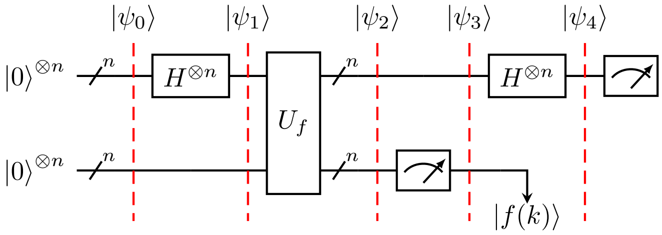
where each alphabet correspond to a unique binary number. So far, none of these values show any similarity which is $2^{n-1}$ query. By evaluating the next data we will figure out $f(100) = c$, then we can say

$$p = 010 \oplus 100 = 110 \tag{3}$$

In the worst-case scenario, one would need $2^{n-1} + 1$ queries to identify $p$, demonstrating the classical inefficiency compared to Simon's quantum approach.

### 1.3 Quantum Solution

While the classical approach to finding the period $p$ involves an exhaustive search that scales poorly with the size of the problem, Simon's algorithm offers a significantly more efficient quantum solution. This quantum algorithm leverages the principles of superposition and entanglement to reduce the number of queries required to determine $p$ from an exponential number to a linear number. Below, we outline the steps involved in Simon's quantum algorithm, demonstrating how it achieves this improvement in efficiency.



It is shown below that the number of trials required to find $p$ is reduced to $O(n)$ if Simon's algorithm is employed. The algorithm is decomposed into the following steps:

1. **Preparation**
   Prepare two sets of $n$-qubit registers in the state

$$|\psi_0\rangle = |0\rangle^{\otimes n} |0\rangle^{\otimes n}, \tag{4}$$

2. **Applying Hadamard Gates to the first register**
   Hadamard gates will apply to the $n$ qubits in the first register to yield

$$|\psi_1\rangle = (H^{\otimes n} \otimes I) |\psi_0\rangle = \frac{1}{2^n} \sum_{x=0}^{N-1} |x\rangle |0\rangle^{\otimes n} \tag{5}$$

3. **Applying Oracle $U_f$**
   Introduce $n$ controlled-NOT gates with control qubits $f_k(x)(1 \le k \le n)$ and the target bit is the $k$th qubit of the second register. We write

$$U_f : |x\rangle |0\rangle^{\otimes n} \to |x\rangle |f(x)\rangle, \tag{6}$$

   where $|f(x)\rangle = |f_1(x)\rangle |f_2(x)\rangle \cdots |f_n(x)\rangle$. Linearity implies the state $|\psi_2\rangle$ after the $U_f$ gate operation on $|\psi_1\rangle$ is

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{N-1} |x\rangle |f(x)\rangle \tag{7}$$

4. **Measuring the second register**
   In fact, we do not need to know the measurement outcome. What we have to do is to project the second register to a certain state $|f(k)\rangle$ for example. After one of these operations, the state is now projected to

$$|\psi_3\rangle = \frac{1}{\sqrt{2}} \Big( |k\rangle + |k \oplus p\rangle \Big) |f(k)\rangle \qquad k \in \{0, 1, \cdots, N-1\} \tag{8}$$

   where we noted that there are exactly two states $|k\rangle$ and $|k \oplus p\rangle$ that give the second register state $|f(k)\rangle$ in step 2.

5. **Applying Hadamard gates**
   Finally we applied again $n$ Hadamard gates to the first $n$ qubits

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2^n}}\sum_{y=0}^{N-1}\Bigg((-1)^{k.y} + (-1)^{(k\oplus p).y}\Bigg)|y\rangle|f(k)\rangle \tag{9}$$

   where

$$\big(k \oplus p\big).y = \big(k.y\big) \oplus \big(p.y\big) \tag{10}$$

   thus

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2^n}}\sum_{y=0}^{N-1}(-1)^{k.y}\Big(1 + (-1)^{p.y}\Big)|y\rangle|f(k)\rangle \tag{11}$$

   for those $y$ that $p.y = 1$, we will not have any contribution because

$$\Big(1 + (-1)^{p.y}\Big) = 0 \tag{12}$$

   so the all contribution belongs to those term that satisfy $p.y = 0$

$$|\psi_4\rangle = \frac{2}{\sqrt{2}}\frac{1}{\sqrt{2^n}}\sum_{\substack{y=0\\ p.y=0}}^{N-1}(-1)^{k.y}|y\rangle|f(k)\rangle \tag{13}$$

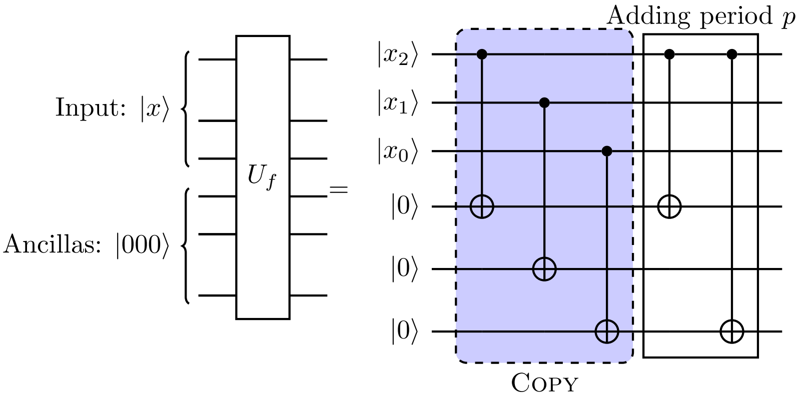
6. **Measurement**
   Finally, we measure the first register, which results in obtaining a state $|y\rangle$ such that $p{\cdot}y = 0$. However, this single equation does not provide enough information to determine the period $p$. Therefore, the algorithm must be repeated multiple times to gather additional measurements, leading to

$$p \cdot y_1 = p \cdot y_2 = \cdots = p \cdot y_m = 0. \tag{14}$$

   It is important to note that at least $n$ iterations are required, as not all equations will be linearly independent. By performing a sufficient number of trials $m$, we can solve Eq. (14) for $p$ using classical methods. The number of trials needed for a successful determination is $O(n)$ with high probability.

## 1.4 Building the Oracle $U_f$

To implement Simon's oracle in a quantum circuit, we must do it differently compared to the Deutsch, Deutsch-Jozsa, and Bernstein-Vazirani algorithms. However, we will maintain our general approach by clarifying everything with an example. Given the content in the next section, we focus on a 3-bit example with period $p = 101$.



To achieve this, we always include a copier block that duplicates the value of the first register $|x\rangle$ into the second register:

$$U_{COPY}\,|x\rangle\,|0\rangle = |x\rangle\,|x\rangle\,. \tag{15}$$

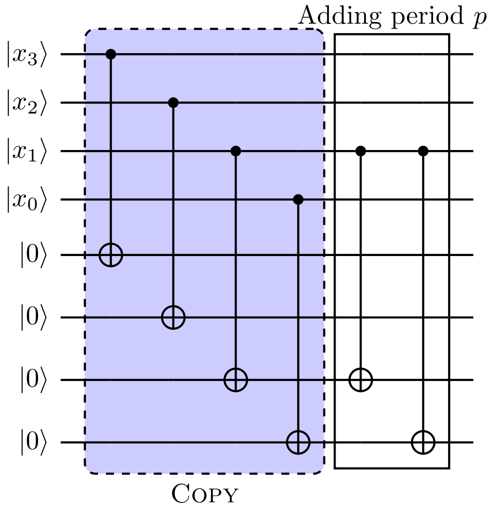
To complete this circuit, we use the period $p = p_2p_1p_0 = 101$. We start from the least significant bit, which is the rightmost digit, and locate the first 1 in the $p$ bit string, here $p_0 = 1$. After identifying the first 1 from the right, we select its corresponding qubit in the first register as a control bit, which is $x_2$ in our case. The final step is choosing the target qubits in the second register according to the number of ones in the period $p$ values, which are the first and third qubits in the second register.

This circuit produces a unique outcome. Let's examine the $f(x)$ values:

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

Surprisingly, we arrive at the same table for Simon's problem but with different $f$ values. Due to the non-sensitivity of Simon's algorithm to the specific values of the $f$ function, we modified them. Thus, we still observe $f(x \oplus y) = f(y)$.

Another example involves a four-qubit system with $p = 1100$:



By illustrating these examples, we emphasize the versatility of constructing Simon's oracle and its invariance to specific function values, which underpins the core principle of Simon's algorithm: detecting hidden periodicities within quantum circuits.

### 1.5 Full Example

Let me consider a function $f : \{0,1\}^3 \to \{0,1\}^3$ which has been defined in Table. (**??**)

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

We can see from this table

$$100 = \begin{cases} f(000) \\ f(101) \end{cases} \quad ; \quad 001 = \begin{cases} f(001) \\ f(100) \end{cases} \quad ; \quad 101 = \begin{cases} f(010) \\ f(111) \end{cases} \quad ; \quad 111 = \begin{cases} f(011) \\ f(110) \end{cases} \tag{16}$$

We can check for each equality $x$ is equal to $x \oplus p$. For finding the period $p$ by using quantum solution we would prepare two registers as

$$|\psi_0\rangle = |000\rangle \otimes |000\rangle \tag{17}$$

in the next step we apply three Hadamard gates to the qubits in first register

$$\begin{aligned} |\psi_1\rangle &= (H^3 \otimes I^{\otimes 3}) |\psi_0\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^{7} |x\rangle \otimes |000\rangle \\ &= \frac{1}{\sqrt{8}} \Big( |000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle \Big) \otimes |000\rangle . \end{aligned} \tag{18}$$

Afterwards we will apply the oracle gate $U_f$ which corresponds to the Table.

$$\begin{aligned} |\psi_2\rangle &= U_f |\psi_1\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^{7} |x\rangle \otimes |f(x)\rangle \\ &= \frac{1}{\sqrt{8}} \Big( |000\rangle |f(000)\rangle + |001\rangle |f(001)\rangle + |010\rangle |f(010)\rangle + |011\rangle |f(011)\rangle + \\ & \quad |100\rangle |f(100)\rangle + |101\rangle |f(101)\rangle + |110\rangle |f(110)\rangle + |111\rangle |f(111)\rangle \Big) \\ &= \frac{1}{\sqrt{8}} \Big( |000\rangle |100\rangle + |001\rangle |001\rangle + |010\rangle |101\rangle + |011\rangle |111\rangle + \\ & \quad |100\rangle |001\rangle + |101\rangle |100\rangle + |110\rangle |111\rangle + |111\rangle |101\rangle \Big) \end{aligned} \tag{19}$$

One can factorize our last equation in the following form

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{8}} \Bigg( \Big( |000\rangle + |101\rangle \Big) \otimes |100\rangle + \Big( |001\rangle + |100\rangle \Big) \otimes |001\rangle + \Big( |010\rangle + |111\rangle \Big) \otimes |101\rangle + \\ & \quad \Big( |011\rangle + |110\rangle \Big) \otimes |111\rangle \Bigg) \end{aligned} \tag{20}$$

the final step is applying three Hadamard gates on the first register

$$|\psi_3\rangle = H^{\otimes 3} |\psi_2\rangle = \frac{1}{8} \sum_{x \in \{0,1\}^{\otimes 3}} \sum_{z \in \{0,1\}^{\otimes 3}} (-1)^{x.z} |z\rangle \otimes |f(x)\rangle \tag{21}$$

But for simplifying this equation we in numerical way, we need to calculate four terms as follows

$$\begin{aligned} H^{\otimes 3}\left(\frac{|000\rangle+|101\rangle}{\sqrt{2}}\right) &= \left(\frac{|+\rangle|+\rangle|+\rangle+|-\rangle|+\rangle|-\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{8}}\frac{2}{\sqrt{2}}\Big(|000\rangle+|010\rangle+|101\rangle+|111\rangle\Big), \end{aligned} \tag{22}$$

$$\begin{aligned} H^{\otimes 3}\left(\frac{|001\rangle+|100\rangle}{\sqrt{2}}\right) &= \left(\frac{|+\rangle|+\rangle|-\rangle+|-\rangle|+\rangle|+\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{8}}\frac{2}{\sqrt{2}}\Big(|000\rangle+|010\rangle-|101\rangle-|111\rangle\Big), \end{aligned} \tag{23}$$

$$\begin{aligned} H^{\otimes 3}\left(\frac{|010\rangle+|111\rangle}{\sqrt{2}}\right) &= \left(\frac{|+\rangle|-\rangle|+\rangle+|-\rangle|-\rangle|-\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{8}}\frac{2}{\sqrt{2}}\Big(|000\rangle-|010\rangle+|101\rangle-|111\rangle\Big), \end{aligned} \tag{24}$$

and the last one will be

$$\begin{aligned} H^{\otimes 3}\left(\frac{|011\rangle+|110\rangle}{\sqrt{2}}\right) &= \left(\frac{|+\rangle|-\rangle|-\rangle+|-\rangle|-\rangle|+\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{8}}\frac{2}{\sqrt{2}}\Big(|000\rangle-|010\rangle-|101\rangle+|111\rangle\Big). \end{aligned} \tag{25}$$

Thus all of these four answer are a superposition of those basis states (000, 010, 101,111) which satisfy $p.y = 0$

$$\begin{aligned} |\psi_3\rangle &= H^{\otimes 3}|\psi_2\rangle \\ &= \frac{2}{8}\Bigg[\Big(|000\rangle+|010\rangle+|101\rangle+|111\rangle\Big)|100\rangle+\Big(|000\rangle+|010\rangle-|101\rangle-|111\rangle\Big)|001\rangle+ \\ &\quad \Big(|000\rangle-|010\rangle+|101\rangle-|111\rangle\Big)|101\rangle+\Big(|000\rangle-|010\rangle-|101\rangle+|111\rangle\Big)|111\rangle\Bigg] \end{aligned} \tag{26}$$

or a little bit simpler as

$$\begin{aligned} |\psi_3\rangle &= \frac{2}{8}\Bigg(|000\rangle\otimes\Big(|100\rangle+|001\rangle+|101\rangle+|111\rangle\Big)+|010\rangle\otimes\Big(|100\rangle+|001\rangle-|101\rangle-|111\rangle\Big)+ \\ &\quad |101\rangle\otimes\Big(|100\rangle-|001\rangle+|101\rangle-|111\rangle\Big)+|111\rangle\otimes\Big(|100\rangle+|001\rangle-|101\rangle-|111\rangle\Big)\Bigg) \end{aligned} \tag{27}$$

When we measure the top output, we will get, with equal probability, 000, 010, 101, or 111. We know that for all these outcomes, the inner product with the missing $p$ is 0. This gives us the set of equations:

$$(000).p_0p_1p_2 = 0 \tag{28}$$
$$(010).p_0p_1p_2 = 0 \tag{29}$$
$$(101).p_0p_1p_2 = 0 \tag{30}$$
$$(111).p_0p_1p_2 = 0 \tag{31}$$

From equation $(000).p_0p_1p_2 = 0$ we cannot find any result for $p$, but for equation $(010).p_0p_1p_2$ we can write

$$\begin{aligned}(010).p_0p_1p_2 &= 0\\ 0 \times p_0 \oplus 1 \times p_1 \oplus 0 \times p_2 &= 0 \rightarrow p_1 = 0.\end{aligned} \tag{32}$$

Beside, from $(101).p_0p_1p_2 = 0$, we will have

$$\begin{aligned}(101).p_0p_1p_2 &= 0\\ 1 \times p_0 \oplus 0 \times p_1 \oplus 1 \times p_2 &= 0 \rightarrow p_0 \oplus p_2 = 0.\end{aligned} \tag{33}$$

since $p_0 \oplus p_2 = 0$, it means they either are zero or one. and finally

$$\begin{aligned}(111).p_0p_1p_2 &= 0\\ 1 \times p_0 \oplus 1 \times p_1 \oplus 1 \times p_2 &= 0 \rightarrow p_0 \oplus p_1 \oplus p_2 = 0,\end{aligned} \tag{34}$$

Now we are ready to find the period $p$. Since, we know $p_1 = 0$ and $p$ cannot be the 000, then it must be $p = 101$ to satisfy $p_0 \oplus p_1 \oplus p_2 = 0$.

## References

[1] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An introduction to quantum computing.* OUP Oxford, 2006.