QIST Project

Hafiz Muhammad Aleem Ullah
Roll no: 22120014
LUMS School of Science and Engineering

Friday, April, 07, 2023

1 Abstract

Simon's algorithm is one of the first quantum algorithms to demonstrate an exponential speedup over classical algorithms for a specific problem. In this report, we provide a comprehensive analysis of Simon's algorithm, including its theoretical background, implementation, and applications. We present the algorithm's complexity analysis and compare it to classical algorithms for solving the same problem. Our findings demonstrate the significant potential of Simon's algorithm in solving complex problems that are difficult to tackle with classical algorithms. We also discuss the quantum circuit implementation for Simon's algorithm. In addition, we present a detailed computational code for implementing this algorithm and analyze its time and space complexity.

2 Introduction

Quantum computing has emerged as a promising paradigm for solving computationally difficult problems in a wide range of fields, including computer science and cryptography. Among the many quantum algorithms that have been proposed, Simon's algorithm stands out for its ability to efficiently solve the hidden subgroup problem, which has important implications for a variety of applications. The hidden subgroup problem (HSP) is a topic of research in mathematics and theoretical computer science. It is a problem of finding a subgroup of a given group that is hidden by a function. The HSP captures problems such as factoring, discrete logarithm, graph isomorphism, and the shortest vector problem.

In this report, we provide a comprehensive analysis of Simon's algorithm, including its theoretical foundations, computational implementation, and experimental results. We begin by introducing the key concepts of quantum computing that underlie the algorithm, including quantum superposition, entanglement, and interference. We then provide a detailed explanation of Simon's algorithm, including its input, output, and the steps involved in its computation. We also discuss the applications of Simon's algorithm in computer science and cryptography, and compare its performance to classical algorithms for solving the same problem. Finally, we present our findings on the experimental implementation of Simon's algorithm, including its limitations and potential future developments. Overall, this report aims to provide a comprehensive understanding of Simon's algorithm and its significance in the field of quantum computing.

3 Mathematical Formulation

3.1 Simon's Problem

We are given an unknown blackbox function $\boldsymbol{f}$, which is guaranteed to be either one-to-one (1 : 1) or two-to-one (2 : 1), where one-to-one and two-to-one functions have the following properties:

One-to-One: Maps exactly one unique output for every input. An example with a function that takes 4 inputs is:

$$f(1) \longrightarrow 1\ ,\ f(2) \longrightarrow 2\ ,\ f(3) \longrightarrow 3\ ,\ f(4) \longrightarrow 4$$

Two-to-One: Maps exactly two inputs to every unique output. An example with a function that takes 4 inputs is:

$$f(1) \longrightarrow 1\ ,\ f(2) \longrightarrow 2\ ,\ f(3) \longrightarrow 1\ ,\ f(4) \longrightarrow 2$$

This two-to-one mapping is according to a hidden bitstring, b , where:

given

$$x_1\ ,\ x_2\ :\ f(x_1)\ =\ f(x_2)$$

it is guaranteed :

$$x_1\ \oplus\ x_2\ =\ b$$

Suppose we have a blackbox $\boldsymbol{f}$, how quickly can we tell if $\boldsymbol{f}$ is one-to-one or two-to-one? Then, if $\boldsymbol{f}$ turns out to be two-to-one, how quickly can we find out b? As it is clear, both cases boil down to the same query of finding b .

3.2 Classical Solution

Classically, if we want to measure b with 100% certainty for a given $\boldsymbol{f}$, we have to check up to $2^{n-1} + 1$ inputs, where n is the number of bits in our input. This means checking over half of all the possible inputs until we find two cases with the same output. We could solve the problem with our first two tries only. But if get an $\boldsymbol{f}$ that is one-to-one, or get really unlucky with an $\boldsymbol{f}$ that's two-to-one, then we have to go with the full $2^{n-1} + 1$ check . There are known algorithms that can do the job in O($2^{n/2}$) checks (Randomized algorithm), but generally speaking the complexity grows exponentially with n.

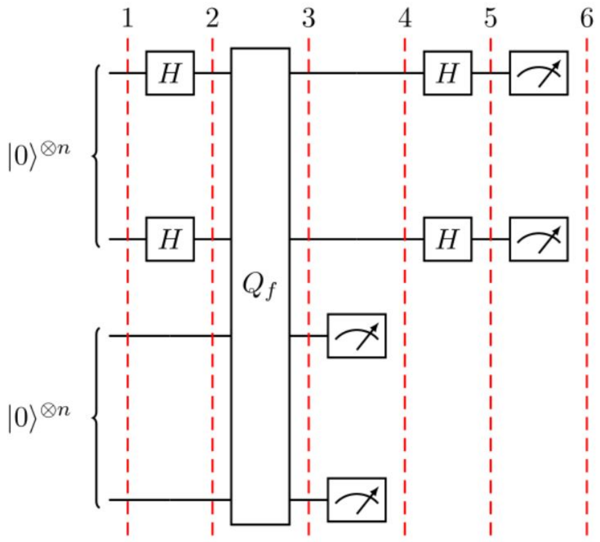

Figure 1: Quantum Circuit

3.3 Quantum Solution

The quantum circuit that implements Simon's algorithm is shown in Figure (1). Where the query function, $\boldsymbol{Q_f}$ acts on two quantum registers as:

$$|x\rangle|a\rangle \;\longrightarrow\; |x\rangle|a \,\oplus\, f(x)\rangle$$

In the specific case that the second register is in the state $|0\rangle = |0\rangle|0\rangle....|0\rangle$ the above equation becomes

$$|x\rangle|0\rangle \;\longrightarrow\; |x\rangle|f(x)\rangle$$

The Simon's algorithm acts in the following steps [1]:

3.3.1 1st step

Two input registers, each of n-qubits length, are initialized to the zero state:

$$|\psi_1\rangle = |0\rangle^{\otimes n}|0\rangle^{\otimes n}$$

3.3.2 2nd step

Hadamard gate applies to the first register:

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

3.3.3 3rd step

Now we apply query function Q_f:

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle$$

3.3.4 4th step

We get a cerain value of $f(x)$ after measuring the second register. Because of the value obtained on the second register, our first register must have some specific related values. If our second register gets $f(x)$ value, first register must possess two values

$$x\ and\ y = x \oplus b$$

Both of these values will be in equal superposition

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}(|x\rangle + |y\rangle)$$

Notice that we have not written here the second register as it is already measured.

3.3.5 5th step

Now Hadamard gate is applied on the first register:

$$|\psi_5\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} [(-1)^{x.z} + (-1)^{y.z}]|z\rangle$$

3.3.6 6th step

Now comes the measurement of first register and it will give us result only if:

$$x.z = y.z$$

$$x.z = (x \oplus b).z$$

$$x.z = x.z \oplus b.z$$

$$b.z = 0(\text{mod } 2)$$

We will get certain value of z whose dot product with is zero. By repeating this algorithm n times, we will get n different values of z. In this way, we can get b (our secret string).

References

[1] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483.