Open Source Implementation of the Durr and Hoyer Algorithm

Mridul Sarkar

University of California-Davis

Chapter 1

Introduction

I began with an idea of quantum neural networks, a seemingly conceivable idea given the naive thought that each node of the classical neural network could be thought of as a particle in space, with quantum mechanics dictating the connections between particles. Though after review of many notable research papers I found the theory was not lacking in this area of quantum computing neither was application. I was amazed by the growth in the field.

Disheartened at first I continued to look for a way to contribute to knowledge in quantum computing. I found Unity Fund and their goals resonated with my own. They proposed a project for an open source package of the Durr and Hoyer Algorithm. It is an algorithm that excites me becasue just a year ago in my first data structures class I was taught O(N) is the best we can do for a list. I was very excited to see an algorithm that pushes this big-O to an awesome $O(\sqrt{N})$.

Chapter 2

Background

2.1 Quantum Computing

Quantum computing's backbone is undoubtedly quantum mechanics. Superposition, Wave Function Collapse, Entanglement, and Uncertainty are utilized by quantum computing to harness the smallest computational unit, a qubit, in order to improve computational effectiveness and efficiency. (1) The details of the quantum mechanic principles used in quantum computing will be explained in section 2.3. Through application of quantum principles we can harness a qubit and from there we can harness multiple qubits by extending quantum principles onto computational space.

2.2 Qubits and Quantum Gates

The classical computer utilizes two states, 0 and 1. Two possible states for a qubit are $|0\rangle$ and $|1\rangle$. Where

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

These states are the orthonormal basis in the complex vector space. It is important to recognize that other states can exist and can be expressed as a linear combination. All properties of a linear combination in real space apply here.(3,9)

$$|\psi| = a_1 |0\rangle + a_2 |1\rangle$$

In order to measure either state $|0\rangle$ or $|1\rangle$ we normalize the coefficients a_1 and a_2 into:

$$|a_1|^2 + |a_2|^2 = 1$$

Measuring the qubits we get either $|0\rangle$ with probability $|a_1|^2$ or $|1\rangle$ with probability $|a_2|^2$. To better under qubits one must examine larger systems that end

up being much more useful in application. There is a more generalized form of ψ. Observe $a = a_1, a_2..., a_n$ as representation of an arbitrary vector.

$$\sum_{i=1}^{n} a_i \left|x_i\right\rangle$$

Following the rules of linear algebra we can extend this definition to apply an arbitrary transformation matrix A which can be understood as a quantum logic gate.
To better understand an arbitrary qubit $\left|x\right\rangle = \left|x_1\right\rangle ... \left|x_n\right\rangle$ we need to understand how the transformation matrix A with i columns and j rows and the arbitrary vector a interact with our n-dimensional qubit. We generalize this to:

$$\psi = \sum_{i=1}^{n} (\sum_{j=1}^{n} A_{ij} a_j) \left|x_i\right\rangle$$

We will now examine some useful Quantum gates. Again quantum gates are analgous to the transformation matrix mentioned above, A. Review of Classical logic gates is recommended.

It is important to remember that any Unitary matrix is a quantum gate. (3)
Not Gate: $\left|0\right\rangle \rightarrow \left|1\right\rangle$
Hadmard Gate:
Pauli-X-Y-Z:
Phase:
pi/8:
Controlled-not:
Swap:
Controlled-Z:
Toffol:
Fredkin:
Measurement: Projection onto $\left|0\right\rangle and \left|1\right\rangle$

2.3 Introduction to Quantum Mechanics and Algorithms

2.3.1 Quantum Mechanics Principles

Schrodinger equation

1-D page 15 (9) time independent (35) 2-D/3-D (135)

Uncertainty principle

page 32,122 of (9)

Identical Particles

page 191 (9)

2.3.2 Quantum Computing Principles

These postulates will help us understand how quantum mechanic principles translate into linear algebra. Allowing us to see numbers and no longer imagine quantum mechanics as some dark magic, as I did about two months ago. Additionally we are given information on how to use tools mathematicians are already familiar with in order to develop our own quantum algorithms.

Postulate 1

Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its state vector, which is a unit vector in the system's state space.

Postulate 2

The evolution of a closed quantum system is described by a unitary transformation. That is, the state—of the system at time t1 is related to the state —of the system at time t2 by a unitary operator U which depends only on the times t1 and t2

Postulate 2'

The time evolution of the state of a closed quantum system is described by the Schrodinger equation

Postulate 3

Quantum measurements are described by a collectionMm of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is—immediately before the measurement then the probability that result m occurs is
" 80-84, (3)

2.3.3 Useful Defintions

Taking the information from 2.3.2 we can understand quantum mechanics and their influence on computing we to help guide us while we build an algorithm. Using these basic definitions we can keep our classically trained brain in check.

Quantum Superposition

If a system can be in state A or state B, it can also being a "mixture" of the two states. If we measure it, we see either A or B, probabilistically. (1) (9)

Quantum Entanglement

If a system can be in state A or B, it has to be a mixture of both states. When measuring the system the independent components cannot be measured unless related to each other. (1) (9)

Wave Function Collapse

When the wave function, existing in the Hermitian space in superposition as multiple eigenstates, collapses to a single eigenstate due to interaction with the external world. (1)(9)

Uncertainty principle

Pairs of measurements where greater certainty of the outcome of one measurement implies greater uncertainty of the outcome of the other measurement.(1)(9)

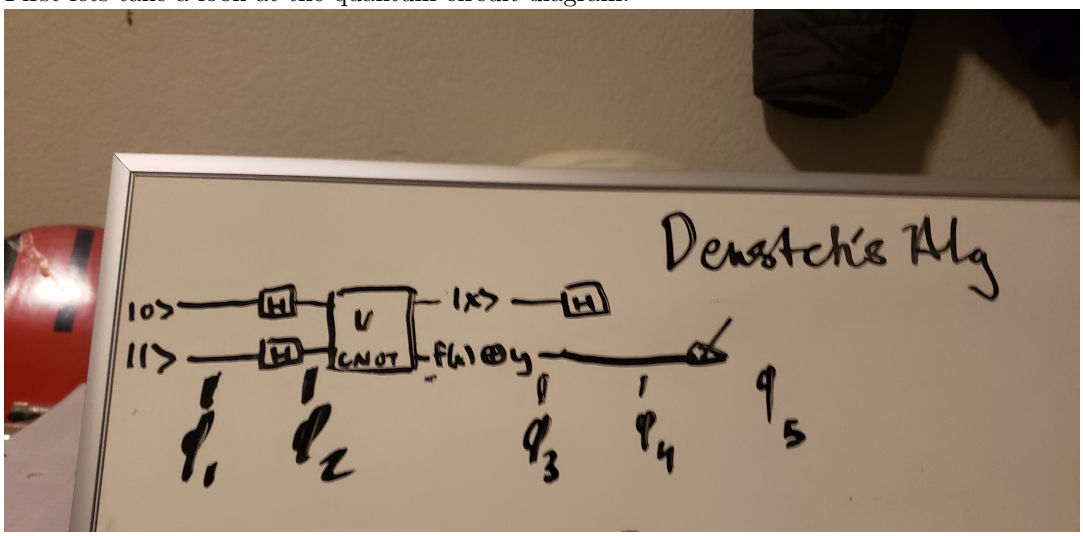
2.3.4 Deutsch's Algorithm

Introduction

This algorithm though not that impressive at first glance showed the scientific community that quantum computing is the next step for computational efficiency. The problem to be solved is as follows: Given f(0)= 0 or 1 and f(1)= 0 or 1 find if f(0) equals f(1) or f(0) does not equal f(1).

Quantum Circuit

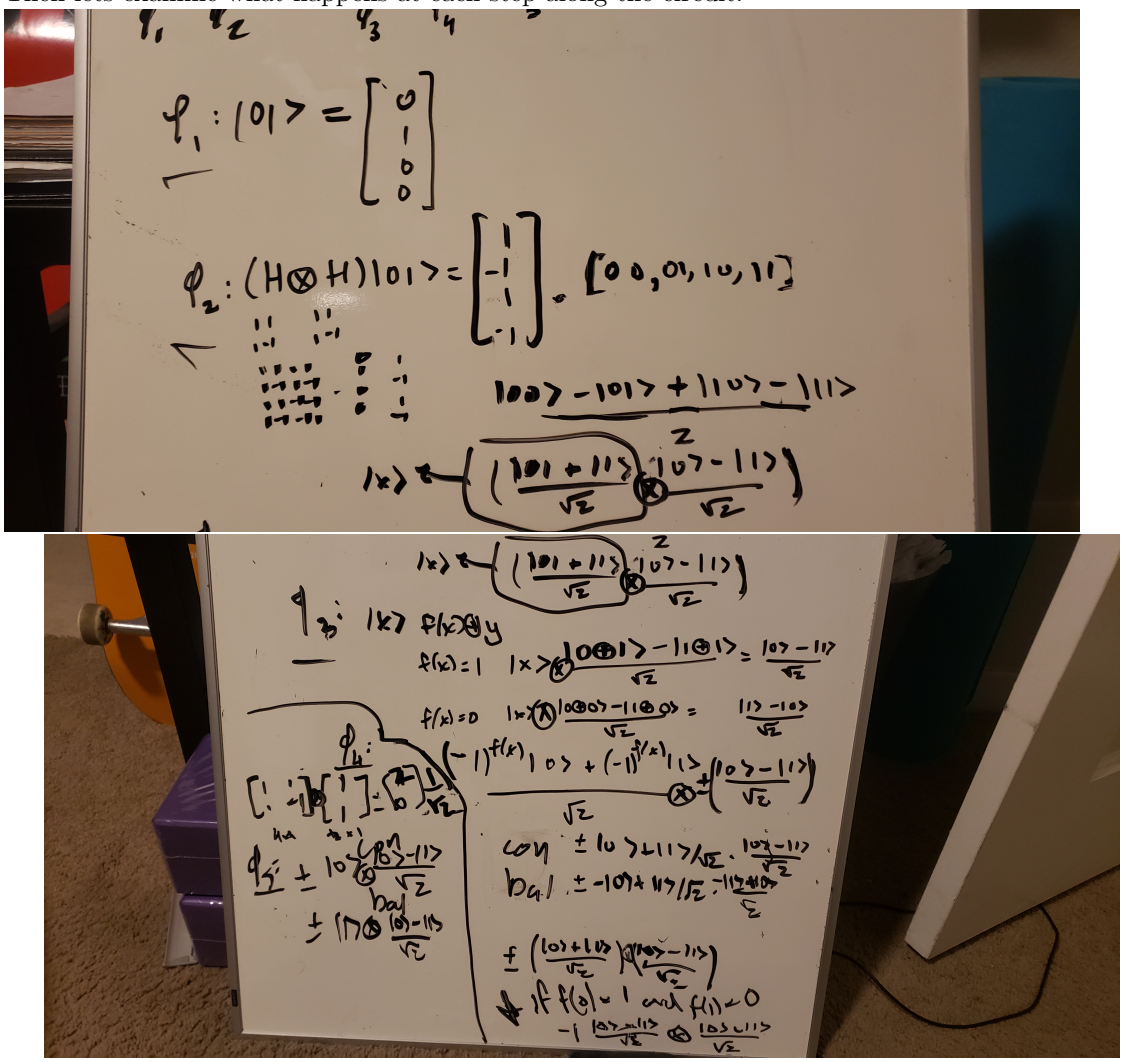
First lets take a look at the quantum circuit diagram:



Hadmard gates are the first step of our algorithm, rotating our tensors $|0\rangle$ $|1\rangle$ 90 degrees resulting in an entangled $|01\rangle$. From here our algorithm utilized CNOT gate, which should be reviewed from the earlier section on quantum gates. The resulting '$|0\rangle$' tensor has the same properties as before the gate. We utilize $|x\rangle$ in order to recognize that is no longer $|0\rangle$. The '$|1\rangle$ tensor is transformed utilizing the $\bigoplus$ operator. The details of this will be explained in the next section. Finally the Hadmard gate is applied once more to the '$|0\rangle$' tensor in order to restore its state and observe the resulting $|1\rangle$ tensor, which will determine whether the function is balanced or constant. This a general overview of the algorithms architecture. It will be useful to reference to this explanation as one begins to understand the algorithm in the next section.

Understanding the Algorithm

Then lets examine what happens at each step along the circuit:



At phi one we show entanglement. at phi 2 we show superposition and uncertainty phi 3 also shows uncertainty and entanglement phi 4 shows us wave function collapse and phi 5 lets us measure our system in order to determine whether it is balanced or constant

Efficiency

Classically this algorithm looks like this:

```
f(0) = 1 or 0
f(1) = 1 or 0

if f(0) = f(1) then constant
if f(0) ~= f(1) then balanced

if f(0) == 1:
    if f(1) == 1:
        constant
    else:
        balanced

else:
    if f(1) == 1:
        balanced
    else:
        constant
```

Two steps must be taken through the classical algorithm in order to determine first if f(0) is 1 or 0 and then to determine what f(1) is. As we saw When Understanding the Algorithm, at the end we have a state that once measured will let us see whether the equation is balanced or constant as we know what value will correspond to the constant or balanced state. Only one step versus two. This not only shows the computational efficiency but the sheer power of quantum principles in computing. Taking a problem that classically would have no room for improving efficiency and beautifully manipulating the problem into a quantum system.

Conclusion and code

2.3.5 Shor's Algorithm

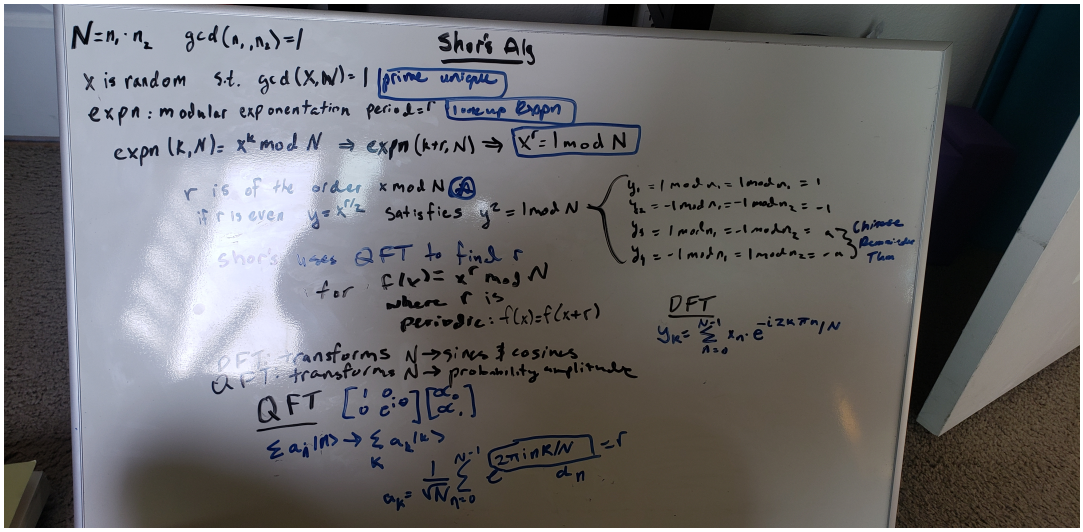
Introduction

At first I wasn't keen on understanding Shor's Algorithm. Truthfully I have never understood cryptography. I can definitely say after pushing myself to understand this algorithm, it is a perfect way to show someone the true power of quantum computing. Duestch's Algorithm let us a very simple and effective rationale for using quantum computers. Shor's Algorithm shakes our classically founded computational world with its power and efficiency. The problem seems simple but is very difficult for classical computers; find all prime factors of a

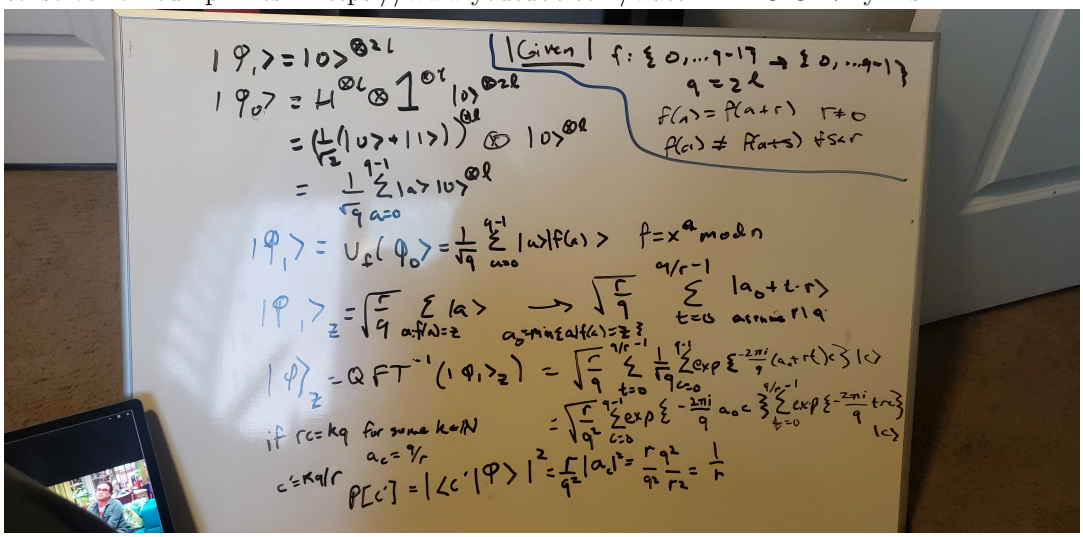
very large number.

Quantum Circuit

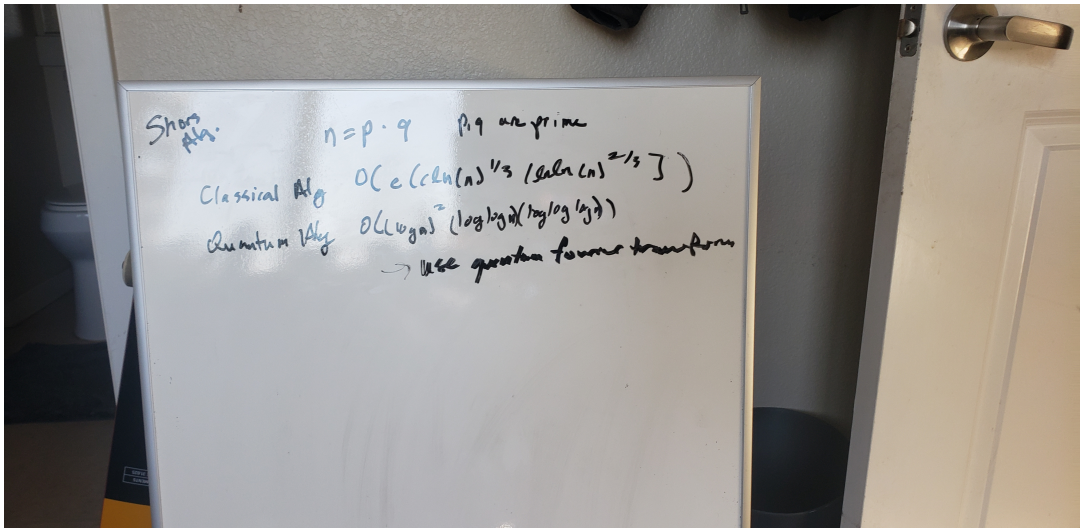
Understanding the Algorithm



Take the number you want to be factored, use number theory to find the period of very long sequence. Then utilize the quantum computer as computational interferometer, using light through diffraction grading, we get a pattern that gives us the space between the different gradings. The period is found through using the quantum interferometer. The final step is to use number theory again to solve for our primes. https://www.youtube.com/watch?v=hOlOY7NyMfs

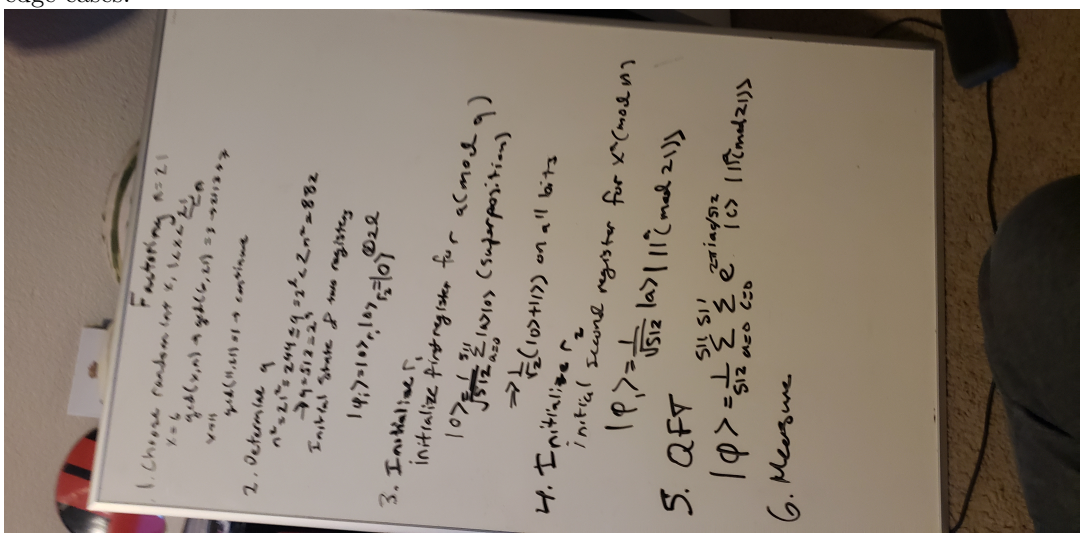


Efficiency



Conclusion

In conclusion we will examine the use of Shors when we have N =21. Lets observe how our algorithms handles factoring primes and observe how to handle edge cases.



2.3.6 Grovers Algorithm

Introduction

Grovers algorithm lets us search through a database unlike any other search algorithm. Using the power of the probabilistic wave function. Entangling and superimposing all the data we are able to effectively determine where our desired data is within our structure, with much more efficiency than any classical search algorithm.

Quantum Circuit

Understanding the Algorithm

Efficiency

Conclusion and Code

]

Chapter 3

Durr and Hoyer Algorithm

3.1 Introduction

3.2 Algorithm

3.2.1 Implementation

3.2.2 Efficiency

3.3 Applications

Chapter 4

Conclusion and Future Work

Chapter 5

Acknowledgments