

**Shor’s Algorithm**

Stratakis Andreas

GitHub repository:

<https://github.com/astratakis/shors-algorithm>

TECHNICAL UNIVERSITY OF CRETE

ELECTRICAL AND COMPUTER ENGINEERING

JULY 2022

**CHAPTER I – ABSTRACT**

Shor’s algorithm is a quantum algorithm for finding prime factors of an integer. The goal of this paper is to explain the problem, the algorithm and to give a generic implementation of the quantum circuit that executes this algorithm.

In number theory integer factorization is the decomposition of a composite number into a product of smaller integers. If these factors are further restricted to prime numbers, the process is called prime factorization.

There is not yet any ‘classical’ algorithm that can factor all integers in polynomial time. That is to factor a bit number in for some constant . Neither the existence nor the non existence of such algorithm has been proven. This problem is currently unsolved in computer science. It is suspected though that no polynomial algorithm exists and thus the problem is in class (although it is not yet prooven). The current best implementation runs in

**CHAPTER I – SHORS ALGORITHM**

procedure shors\_algorithm(N) -> [p, q]:

# Start with a bad guess

a = random(2, N-1)

**CHAPTER II – QUANTUM FOURIER TRANSFORM (QFT)**

In quantum computing the Quantum Fourier Transform (QFT) is a linear transformation on qubits, and is the quantum analogue of the discrete Fourier transform. The QFT applies on the amplitudes of a wavefunction and produces a different wavefunction as shown below.

The discrete Fourier transform acts on a vector and maps it to another vector according to the formula:

Similarly, the quantum Fourier transform acts on a quantum state and maps it to the quantum state where:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Note that only the amplitudes of the state were affected by this transformation. The QFT can also be expressed as the unitary matrix:

, where

The QFT transforms between two basis, the computational (Z) basis and the Fourier basis. Every possible multi-qubit state in the computational basis has a distinct corresponding state in the Fourier basis. Note that N refers to the number of dimensions in the Hilbert space of qubits, thus , where is the number of qubits.

A more intuitive way to understand the quantum Fourier transform is the following. Let’s assume a basis state in the Z axis of the Hilbert space of qubits. The transformation will be:

, where

Notice that is a phase rotation on a particular state. The probability of measuring each state does not change because:

**Example 1:**

Let’s assume that we have 1 qubit. This means that and

Notice that the QFT on 1 qubit is equal to the Hadamard transformation.

**Example 2:**

This is another example where qubits. This means that and . In this case the unitary QFT matrix is:

The interesting thing is the general effect of the QFT in the initial state.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Example 3:

A more interesting example is one where we have 5 qubits. In this case the unitary matrix cannot be displayed because .

**CHAPTER III – QUANTUM FOURIER TRANSFORM (QFT)**