Minor Project
On

Grover's Search Algorithm on a Quantum Computer

by

Saasha Joshi
UE173085
BE CSE (7th Semester)

Mentor
Dr. Deepti Gupta

Computer Science and Engineering
University Institute of Engineering and Technology
Panjab University, Chandigarh-160014, INDIA

2020

Abstract

Quantum computing is a quickly growing field of research thanks to recent hardware advances. This project provides an analysis of a quantum algorithm that searches an element in an unsorted set of numbers faster than any classical search algorithm. Starting by discussing the basic concepts of qubits in a Quantum Computer, full implementation of Grover's Search Algorithm in a search space of 4 qubits is presented on a real Quantum Computer (IBM).

In addition to this, the project discusses the computational complexity of Grover's Algorithm and concludes by comparing the theoretical and real-life execution results of the algorithm. Further applications of Grover's Algorithm, for example, in solving a Sudoku problem, Max-Cut Problem, Lights-Out Problem, Vertex Cover problem, and Quantum RAM implementation are also explored.

TABLE OF CONTENTS

Introduction

Grover's Algorithm is a quantum algorithm that conducts a search on an unsorted dataset with N entries in O(√N) time and O(logN) storage space [1]. Classically, performing a linear search on an unsorted dataset takes O(N) time in the worst-case scenario i.e. using brute-force methods. However, Grover's Algorithm implemented on a quantum computer provides a quadratic speed up to the same problem.

Where a classical computer uses binary bits to store its information, a quantum computer uses quantum bits, called qubits. This unique quantum characteristic of qubits allows quantum computers to solve a certain class of problems faster than classical computers.

IBM has several quantum computers available to the public through their cloud service IBM Q. The ibmqx2 is a superconductivity-based 5-qubit quantum computer available through a Python-based programming interface called Qiskit. The interface also provides access to an ibmqx2 simulator where simulation runs can be performed.

1.1. Topics Discussed

Quantum Computing, Qubits, Grover's Algorithm, Computational Complexity, Oracle, Amplitude Amplification, Theoretical Results, Execution Results.

1.2. Purpose and Scope

The purpose of this project is to provide a detailed analysis of Grover's Algorithm for unsorted search in a search space of 4 qubits. The algorithm is implemented using the Qiskit SDK and involves the use of a programmable IBM Quantum Computer (ibmqx2) with the help of available quantum gates. The project further explores the real-life applications of Grover's Algorithm, for example, in solving a Sudoku problem, Max-Cut Problem, Lights-Out Problem, and Quantum RAM implementation

Background

Suppose we are given a large list of N items. Among these items there is one item that we wish to locate; we will call this one the winner (w). Let us say all items in the list are gray except the winner (w), which is highlighted below.

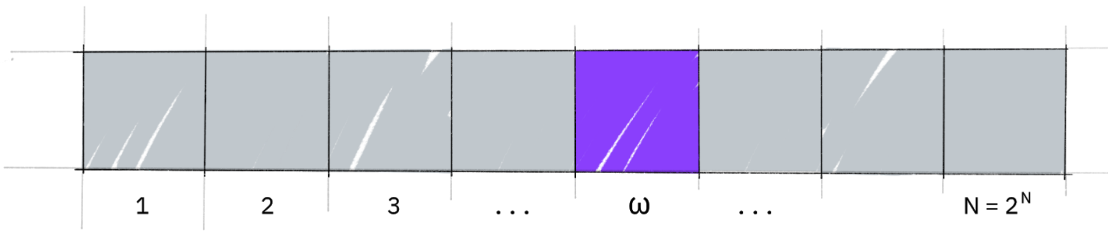


Figure 1. Marked item (w) in a dataset

To find the highlighted box -- the marked item -- using classical computation, we would have to check on average N/2 of these boxes, and in the worst case, all N of them. On a quantum computer, however, we can find the marked item is roughly √N steps with Grover's amplitude amplification trick. A quadratic speedup is indeed a substantial time-saver for finding marked items in long lists [2].

Thus, given an unordered set of N = 2n states X = {x1, x2, ..., xN } and a binary function f : {0, 1} the algorithm will find a state x' such that f(x') = 1.

2.1. Stages of the algorithm

The algorithm is divided into the following four stages.

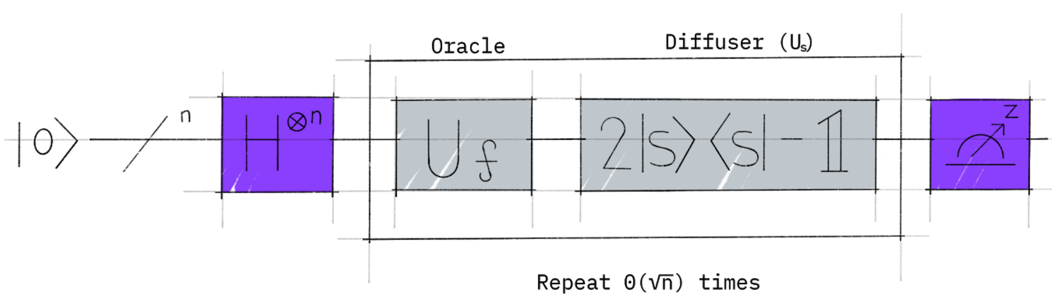


Figure 2. Stages of Grover's Algorithm

Initialization

All qubits are set to be in superposition. All states have the amplitude $1/\sqrt{N}$.

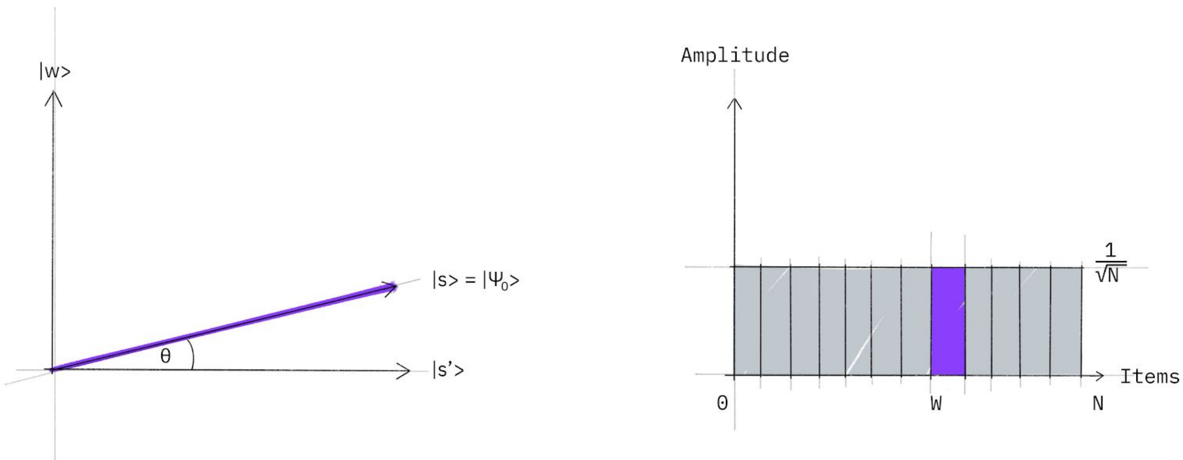


Figure 3. Step 1: Initialization of the qubits in a uniform superposition |s⟩

Oracle

The oracle function marks the state x that satisfies the condition f(x) = 1 by performing a phase flip. All other states are left unaltered. The operation has the effect of inverting the state's amplitude and it runs in constant time.
Grover's algorithm solves oracles that add a negative phase to the solution states. I.e. for any state $|x\rangle$ in the computational basis:

$$U|x\rangle = \begin{cases} |x\rangle & \text{if } x \neq \omega \\ -|x\rangle & \text{if } x = \omega \end{cases}$$

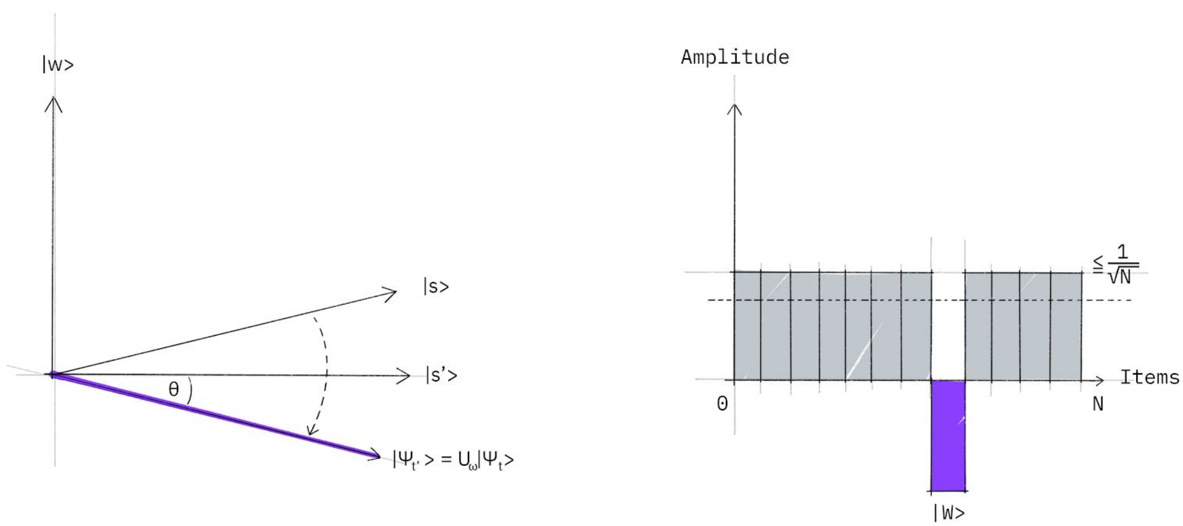
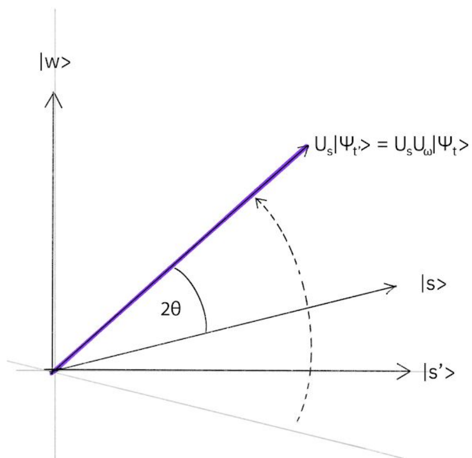


Figure 4. Step 2: Application of the oracle (Uf) for reflection of the state |s⟩ about |s′⟩

Amplification

The amplification stage phase flips the amplitudes around the average amplitude. As the target state’s amplitude was inverted while the other states kept their original amplitudes, the flip causes the target state’s amplitude to increase and the others to decrease.



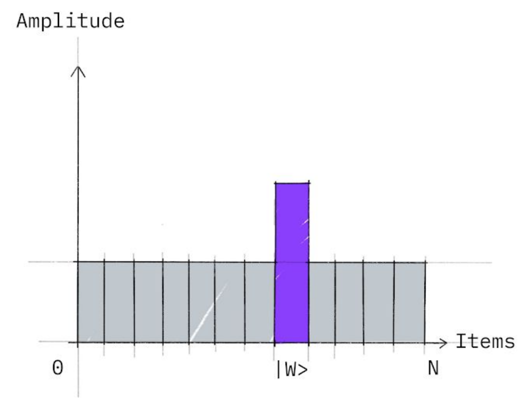


Figure 5. Step 3: Amplitude Amplification by an additional reflection (Us) of the new state about the state |s⟩

Measurement

The qubits are read and the output is given.

Design of Algorithm

Initialization

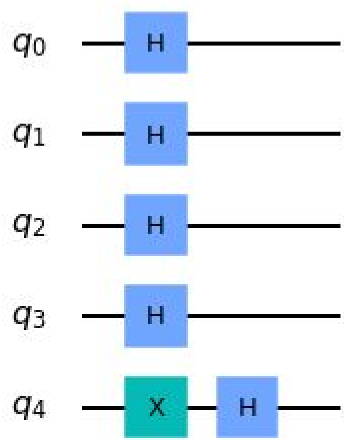


Figure 6. Initialization of 4 qubits and auxiliary qubit

Oracle

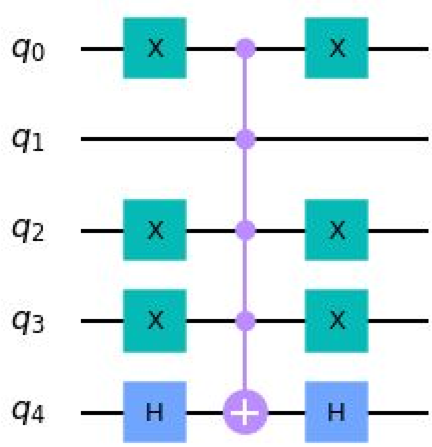


Figure 7. Oracle for w = 0010 i.e. target number 2

Amplification

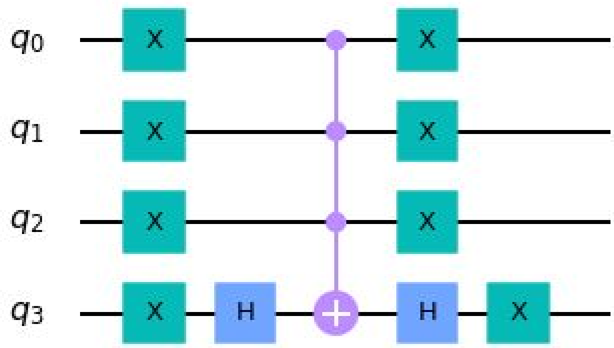


Figure 8. Diffuser for amplitude amplification

Measurement

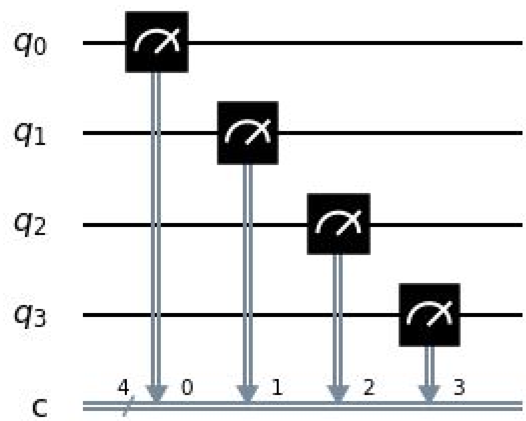


Figure 9. Measurement of the qubits

Implemented Circuit

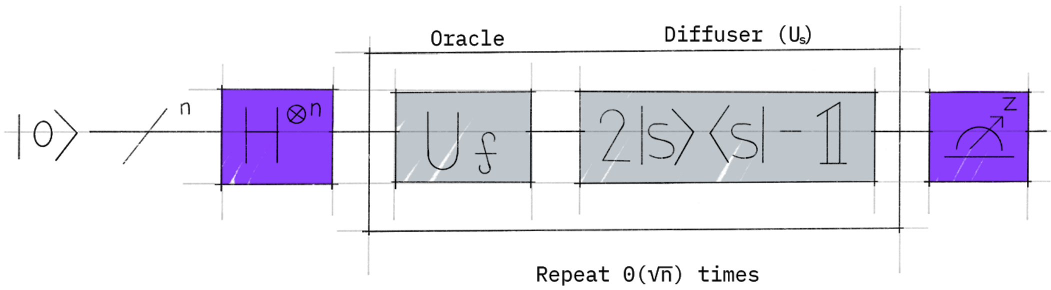


Figure 10. Design of Circuit

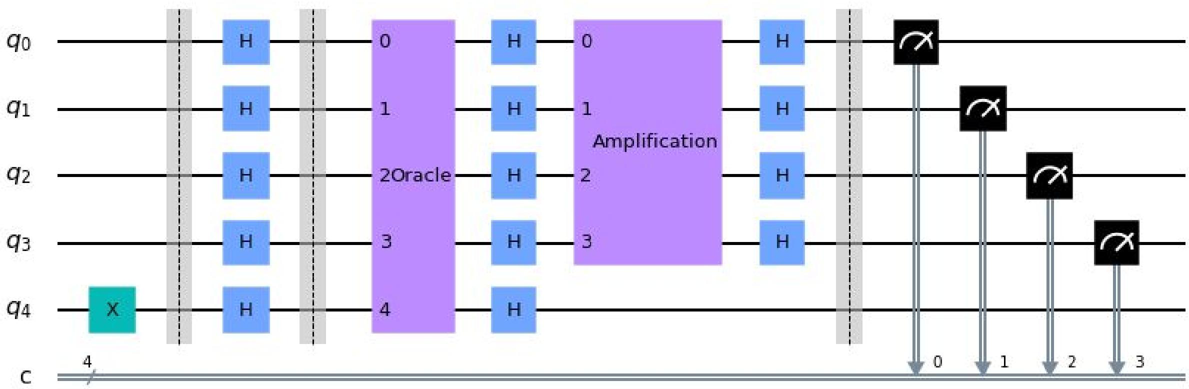


Figure 11. Implemented Circuit of Grover's Algorithm

Hardware and Software Specifications

Hardware Requirements:

Processor	IBM Quantum Backend (ibmqx2)
Qubit Count	5 qubits
Classical Bit Count	4 bits
QRAM	Yes, can be implemented

Software Requirements:

Operating System	Linux / Windows
Development Environment	Jupyter Notebook, Google Colab
SDK / Libraries	Python, Qiskit SDK, Python Libraries (Pandas, NumPy, Matplotlib, Seaborn)

Results

(Changes Required)

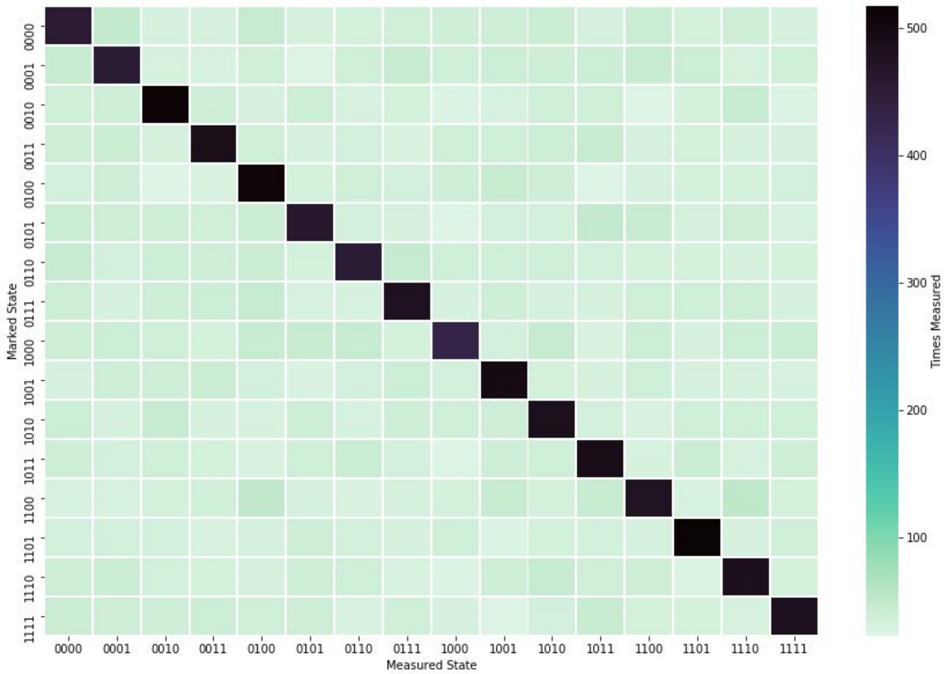


Figure 12: Quantum Simulator Results (Marked vs Measured State)

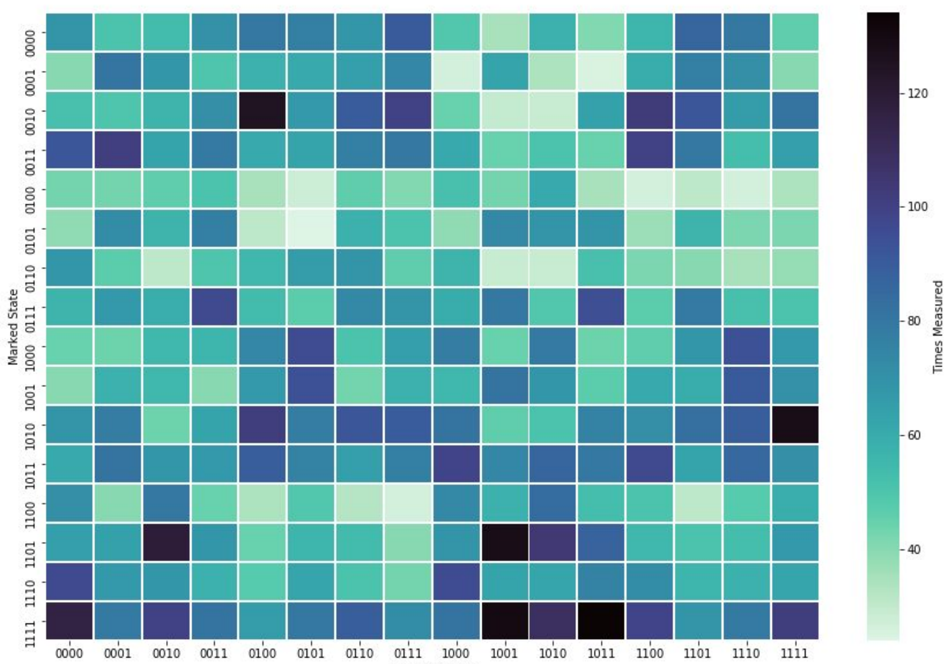


Figure 13: Quantum Backend (ibmqx2) Results (Marked vs Measured State)

Progress

- **September 2020 -** Study Grover's Algorithm
  - Introduction to Quantum Computing Done
  - Introduction to Grover's Algorithm Done
- **October 2020 -** Implement Grover's Algorithm with Qiskit SDK
  - Circuit Implementation (4 stages) Done
- **November 2020 -** Analysis and Compilation of the results
  - Quantum Simulator Results Done
  - Quantum Backend (ibmqx2) Results Done
  - Plotting of Results Changes Required
- **December 2020 -** Explore applications of Grover's Algorithm
  - Sudoku Problem
  - Vertex Cover Problem Done
  - Lights-Out Problem Done
  - Max-Cut Problem Done
  - QRAM Done

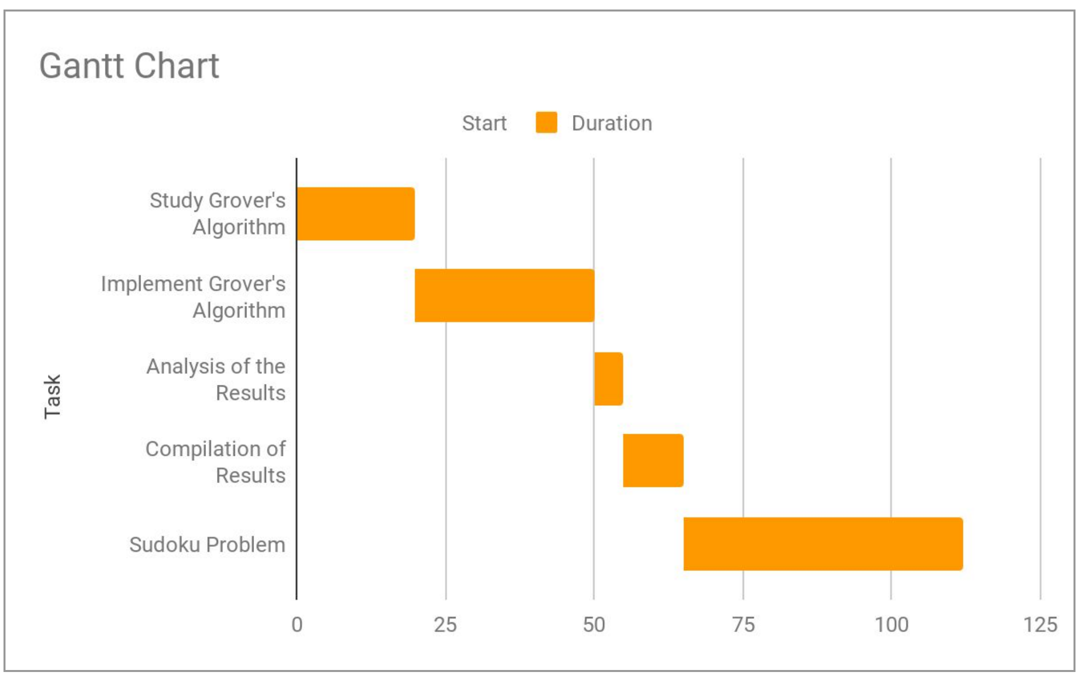


Figure 14: Progress Gantt Chart

References

1. Grover LK. A fast quantum mechanical algorithm for database search. InProceedings of the twenty-eighth annual ACM symposium on Theory of computing 1996 Jul 1 (pp. 212-219).
2. https://qiskit.org/textbook/ch-algorithms/grover.html
3. Figgatt C, Maslov D, Landsman KA, Linke NM, Debnath S, Monroe C. Complete 3-Qubit Grover search on a programmable quantum computer. Nature communications. 2017 Dec 4;8(1):1-9.
4. Strömberg P, Blomkvist Karlsson V. 4-qubit Grover's algorithm implemented for the ibmqx5 architecture.