GROVER'S ALGORITHM

in a 4-qubit Search Space

WHAT IS A QUANTUM COMPUTER?

A classical computer uses bits to store information in the form of binary, 0 and 1.

A quantum computer, on the other hand, uses quantum bits called qubits. These two-level quantum systems display unique physical characteristics that allow them to represent different states at the same time in the form of a linear combination of all.

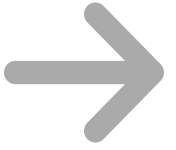
This unique quantum characteristic of qubits allows quantum computers to solve a certain class of problems faster than classical computers.

WHAT IS GROVER'S ALGORITHM?

Grover's Algorithm is a quantum algorithm that conducts a search on an unsorted dataset with N entries in $O(\sqrt{N})$ time and O(logN) storage space.

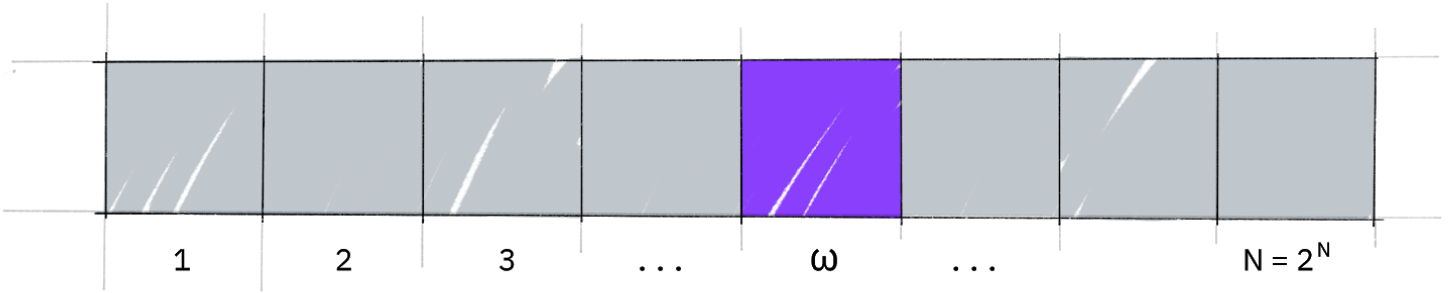
WHAT HAPPENS CLASSICALY?

A classical computer can find an element from an unsorted database, of size N, with an average of N/2 brute-force queries i.e. in O(N) time.



BACKGROUND

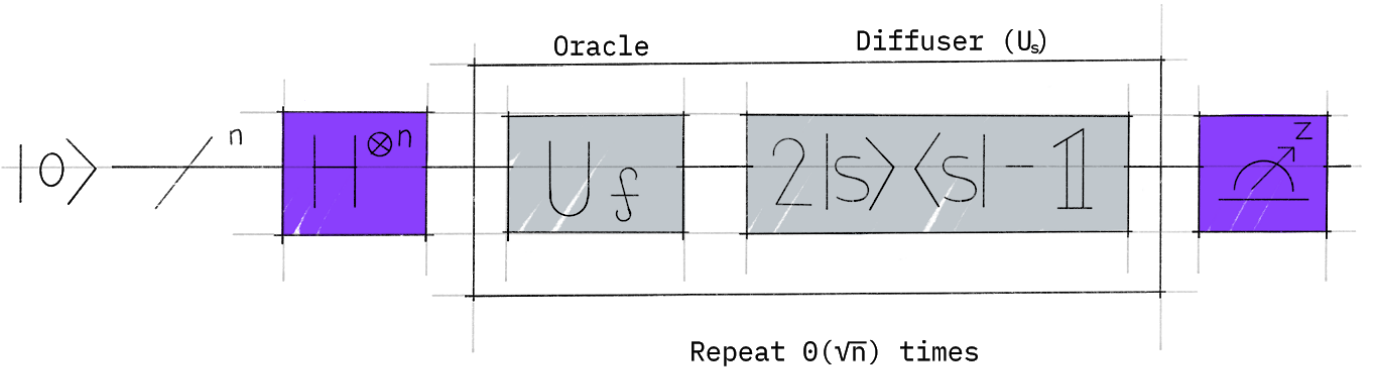
Suppose we are given a dataset of N items. Among the items in the dataset, there is one item that we wish to locate; we will call this one the target number (w).



To find the highlighted box, the marked item (w), using classical computation, we would have to check on average $N/2$ of these boxes, and in the worst case, all N of them.

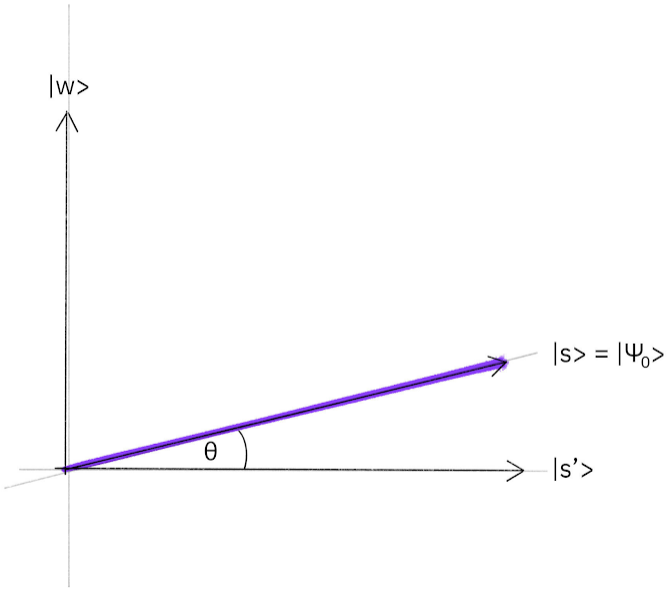
On a quantum computer, however, we can find the marked item is roughly $\sqrt{N}$ steps with Grover's amplitude amplification trick.

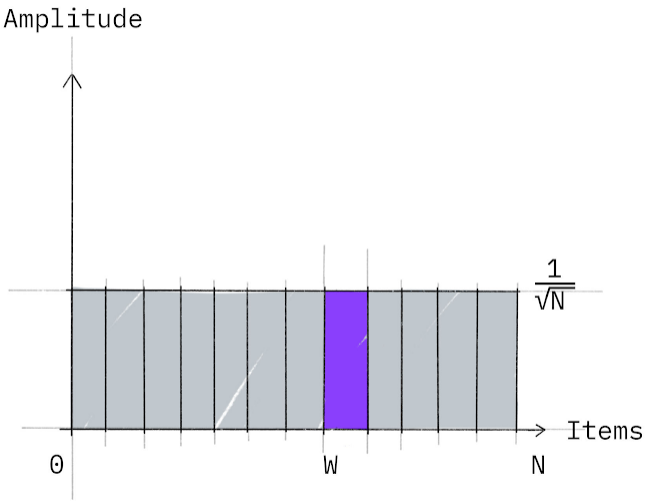
STAGES OF GROVER'S ALGORITHM



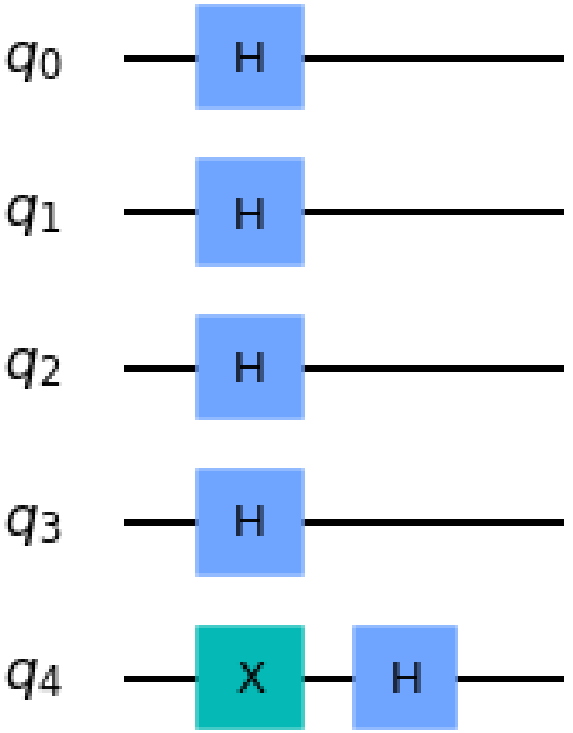
INITIALIZATION

The initialization stage of the algorithm takes the initial quantum state and sets it into a state of equal superposition.





IMPLEMENTATION




```python
def initialize(circuit, n):
    circuit.x(n)
    circuit.barrier()

    circuit.h(range(n+1))
    circuit.barrier()
```

ORACLE - BOOLEAN ORACLE

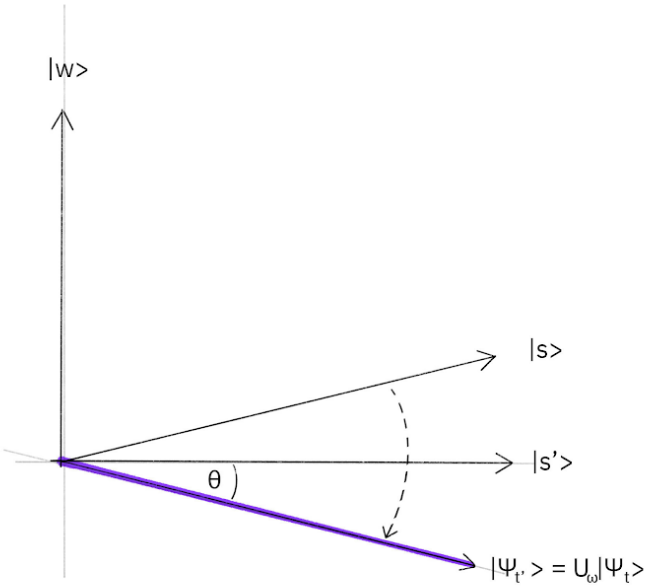
The Oracle stage in the algorithm marks the target state (w) which satisfies the condition,

$$f(w) = 1$$

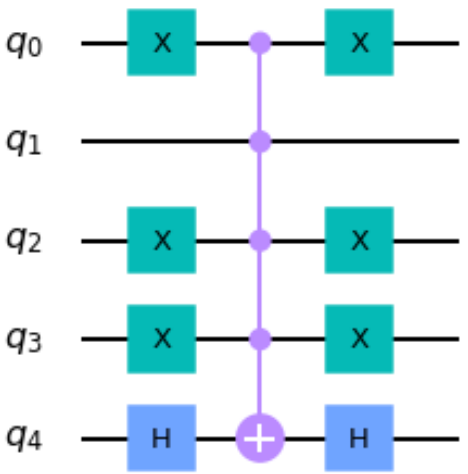
This operation has the effect of inverting the amplitude of the target state while adding a negative phase to the overall quantum state.

$$x = (-1)^{f(x)} x$$

ORACLE - BOOLEAN ORACLE



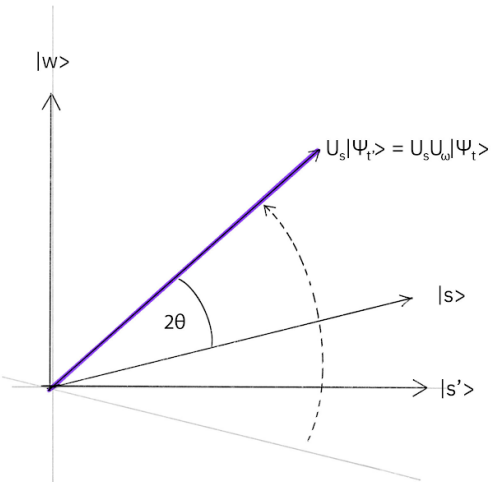
IMPLEMENTATION

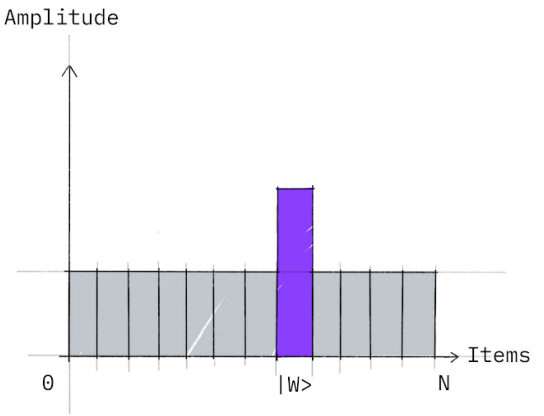



```python
def grover_oracle(n, target_binary):
    circuit = QuantumCircuit(n+1)

    # Grover's Oracle
    for index, value in reversed(list(enumerate(target_binary
        if value == '0':
            circuit.x(n-1-index)

    control_qubits = [0, 1, 2, 3]
    circuit.mct(control_qubits, n, n)

    for index, value in enumerate(target_binary):
        if value == '0':
            circuit.x(n-1-index)

    oracle_gate = circuit.to_gate()
    oracle_gate.name = 'Oracle'

    return oracle_gate
```

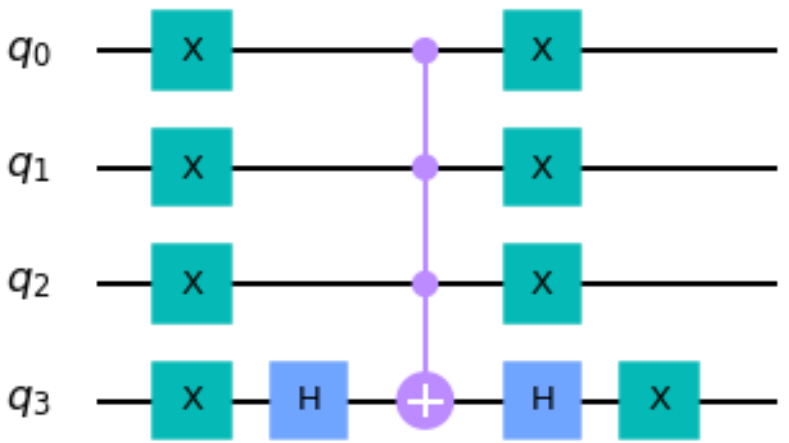
AMPLITUDE AMPLIFICATION

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.





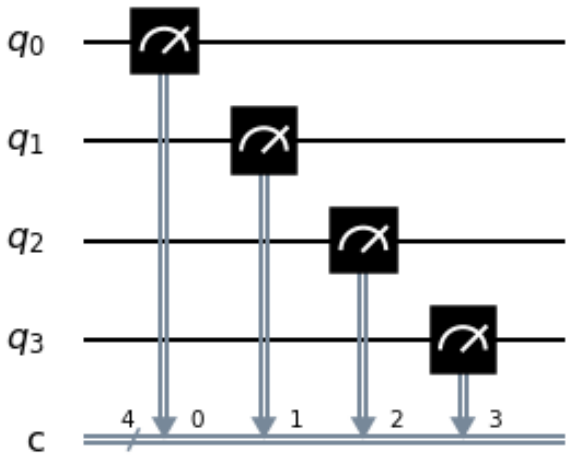
IMPLEMENTATION




```python
def amplification(n):
    circuit = QuantumCircuit(n)

    circuit.x(range(n))

    # Amplifier
    # V
    circuit.cu1(pi/4, 0, 3)

    # V-dagger
    circuit.cx(0, 1)
    circuit.cu1(-pi/4, 1, 3)
    circuit.cx(0, 1)

    # V
    circuit.cu1(pi/4, 1, 3)

    # V-dagger
    circuit.cx(1, 2)
    circuit.cu1(-pi/4, 2, 3)

    # V
    circuit.cx(0, 2)
    circuit.cu1(pi/4, 2, 3)
```

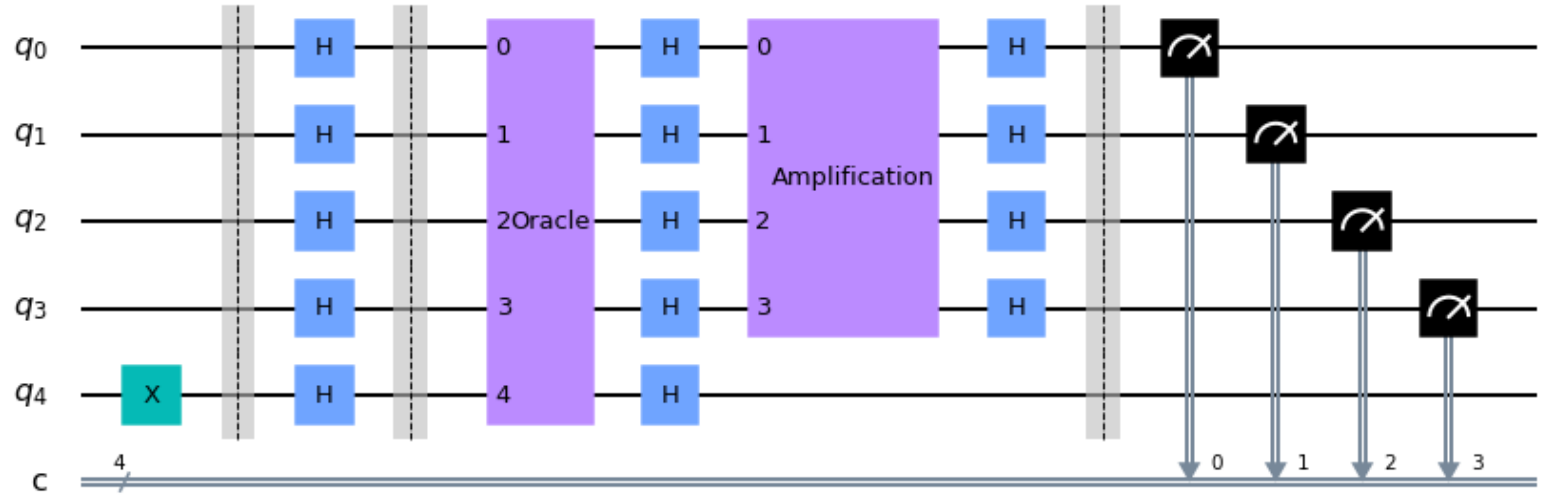
MEASUREMENT

The measurement stage of the algorithm measures the final output of the circuit. After the measurement, the qubits are no longer in superposition as they tend to lose all their quantum physical properties and collapse to give an outcome of one of their possible states.



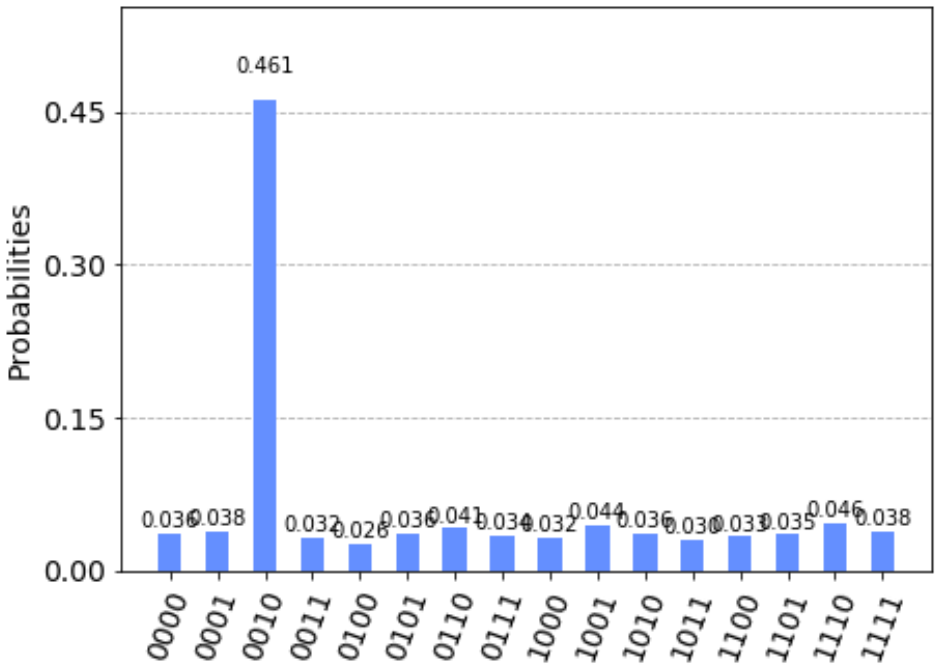

```
1 def measurement_qc(circuit, n):
2   circuit.measure(range(n), range(n))
3   circuit.draw('mpl')
```

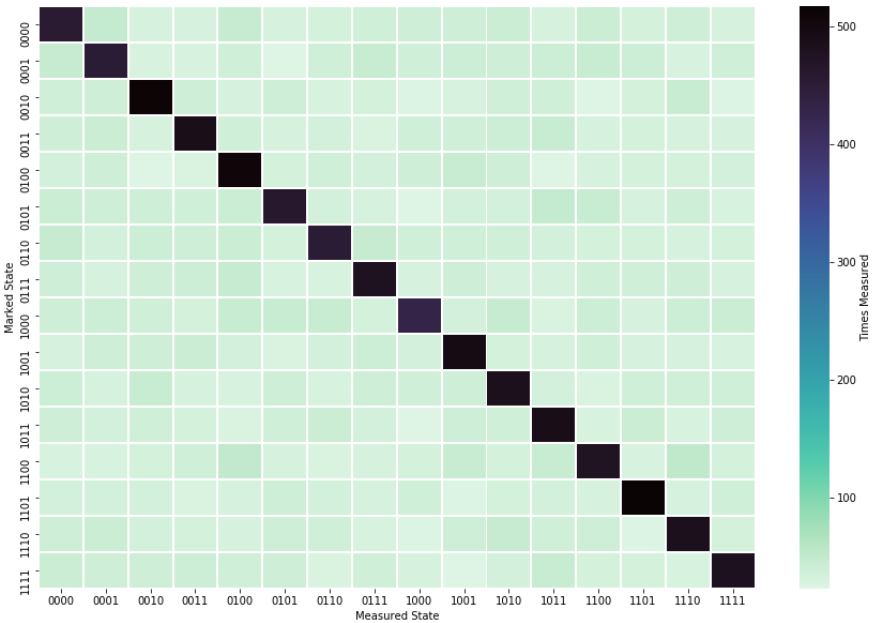
IMPLEMENTED CIRCUIT



RESULTS

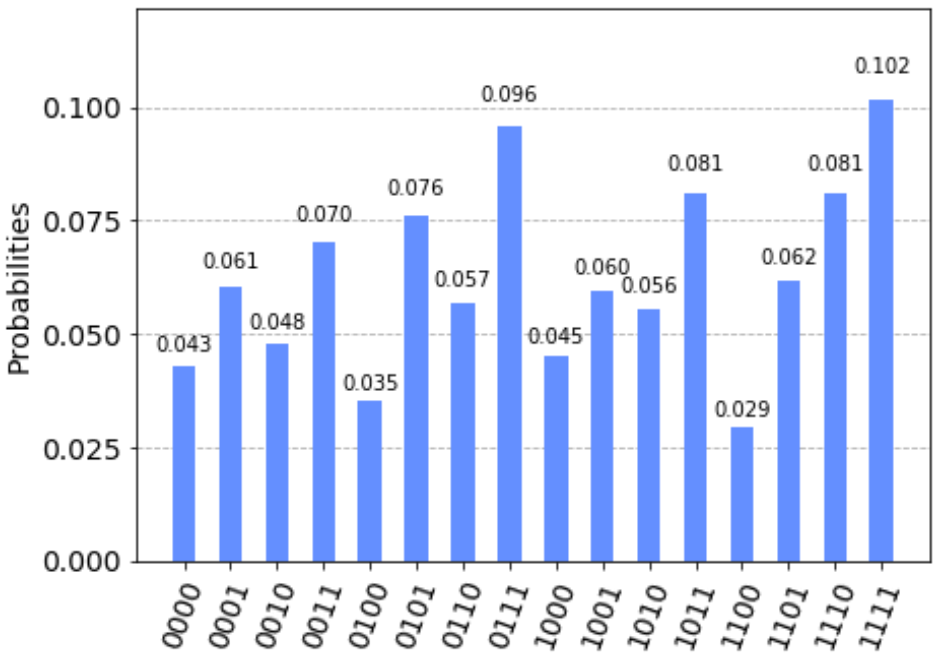
QUANTUM SIMULATOR

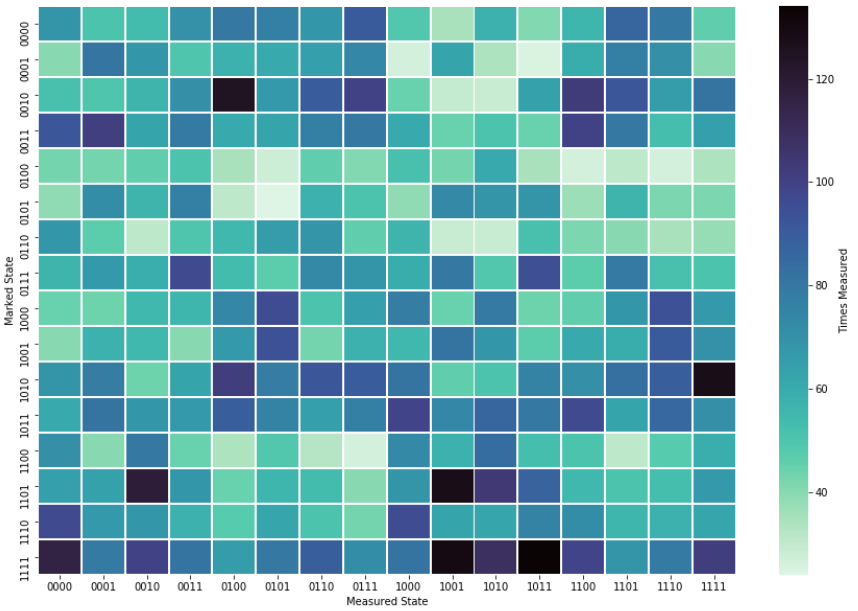




RESULTS

QUANTUM BACKEND (IBMQX2)



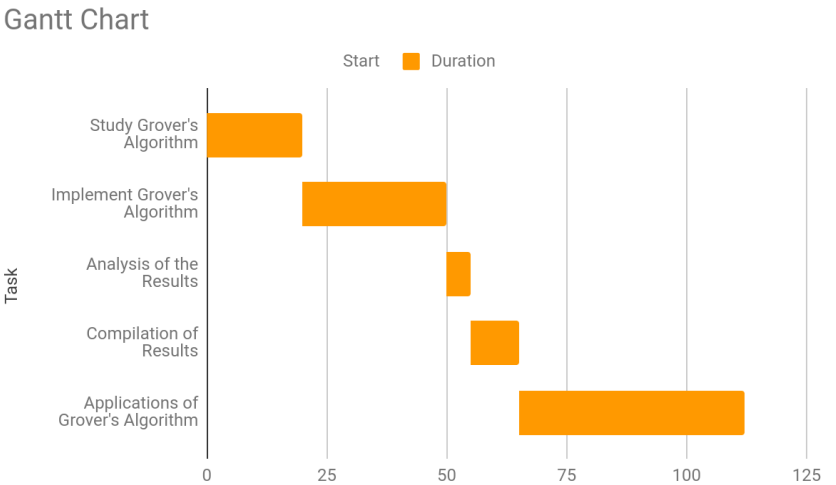


RESULTS

Backend	Number of Qubits	Number of Shots	Algorithm Success Probability (%)	Average Runtime (s)
Qasm Simulator	4	1024	46.92	1.30
Qasm Simulator	4	8192	47.28	1.32
IBMQ Santiago	4	1024	5.33	17.16
IBMQ Santiago	4	8192	5.31	19.54

PROGRESS

- September 2020
  - Introduction to Quantum Computing
  - Introduction to Grover's Algorithm
- October 2020
  - Circuit Implementation with Qiskit SDK
- November 2020
  - Analysis and Compilation of the results
  - Plotting of Results
- December 2020
  - Explore applications of Grover's Algorithm
    - Sudoku Problem, Vertex Cover Problem Done, Lights-Out Problem, Max-Cut Problem, QRAM



ACHIEVEMENTS

- Participated in the IBM Quantum Challenge (Fall 2020) and earned a position in the top 150 among 2500 people. Received an Advanced Badge in Quantum Computing for the same. Badge Link

- Offered an internship at Tech Mahindra from January 2020 to June 2020 as a Quantum Computing Trainee.

SUBMITTED BY: SAASHA JOSHI

MENTOR: DR. DEEPTI GUPTA