Quantum Pattern Matching

201601065 - Samhitha Gundam
201601129 - Kritika Gupta
201601401 - Mit Vasani
201601412 - Khanti Rindani
201601420 - Dhvanee Mehta

Approaches

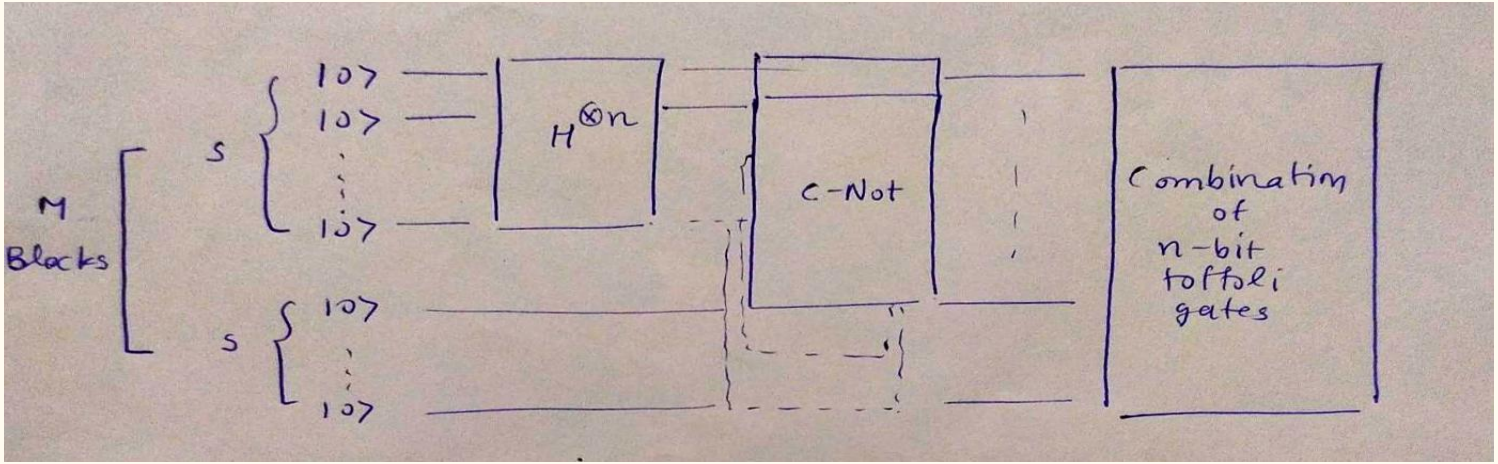
Approach 1: Exact Pattern Matching

- Initialization
- Mark the exact pattern “p” through the Oracle
- Grover Search
- Amplitude Amplification
- Measurement
- Results

Approach 2: Closest Pattern Matching

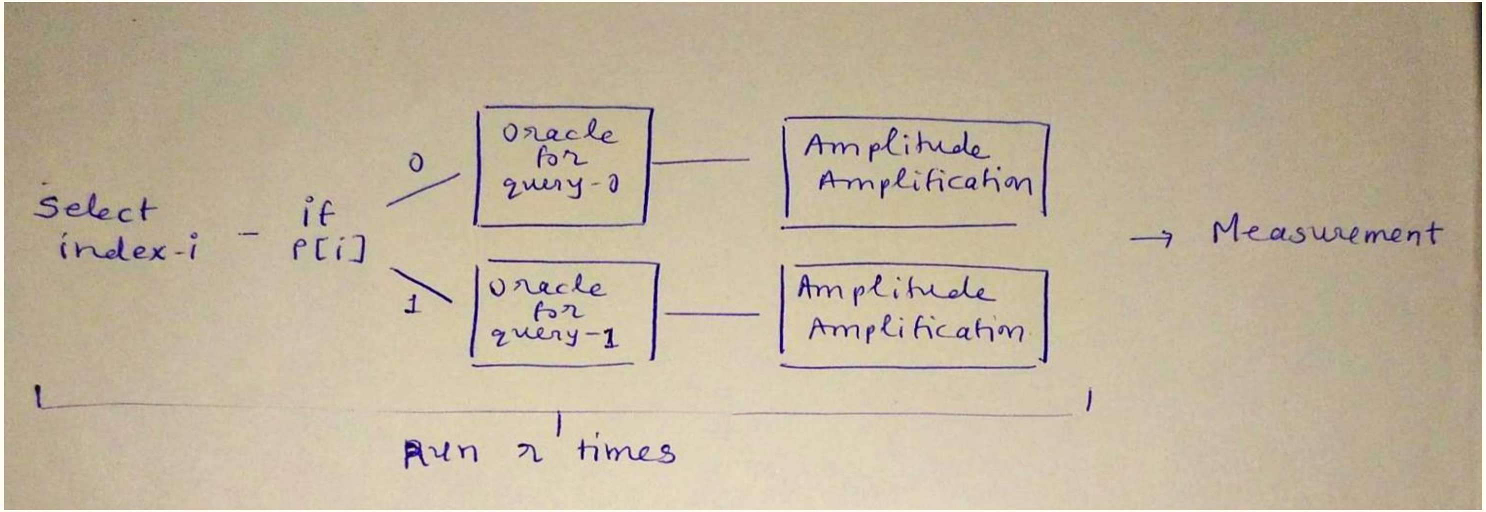
Initialization

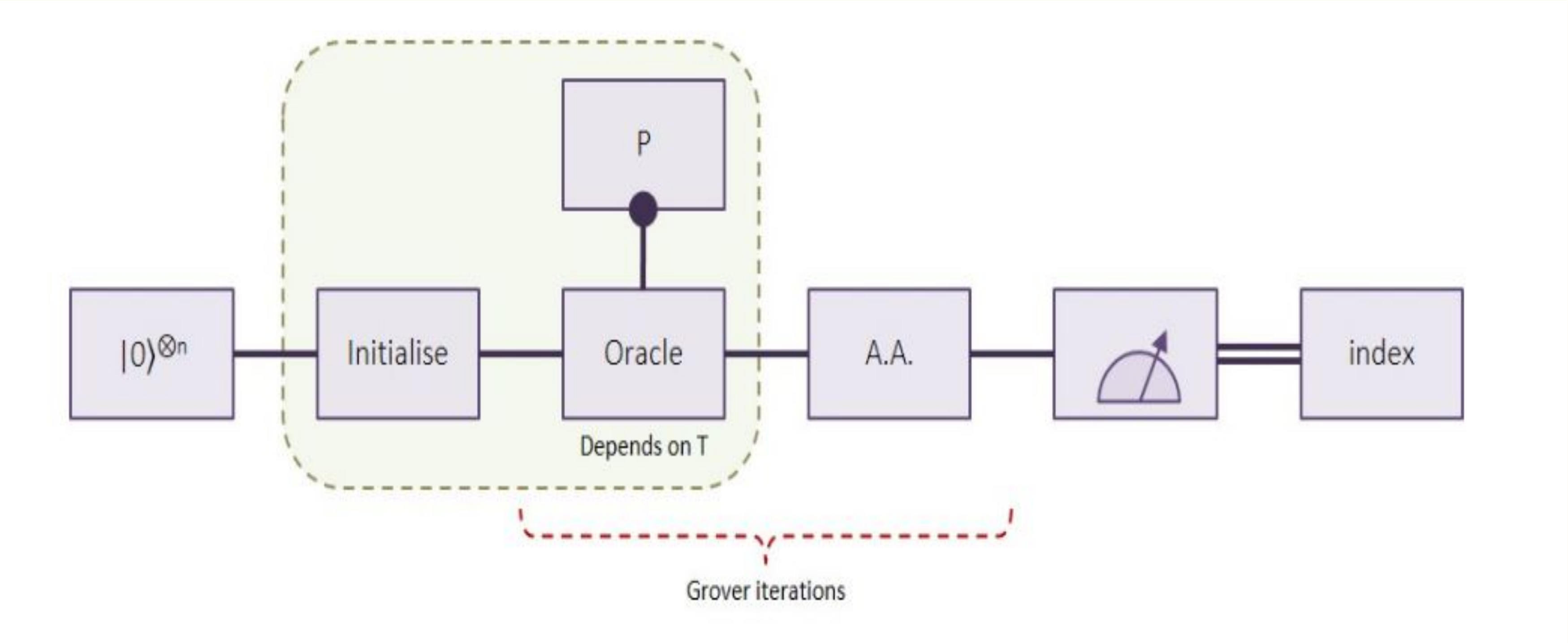
$$|\psi_0\rangle = \frac{1}{\sqrt{N-M+1}} \sum_{i=0}^{N-M} |i, i+1, \cdots, i+M-1\rangle$$



Modified Grover Search

- Oracle marking
- Amplitude Amplification

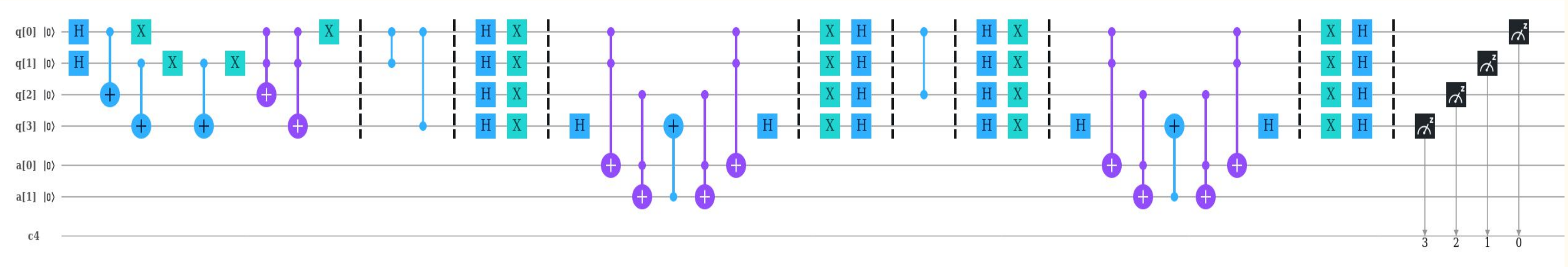


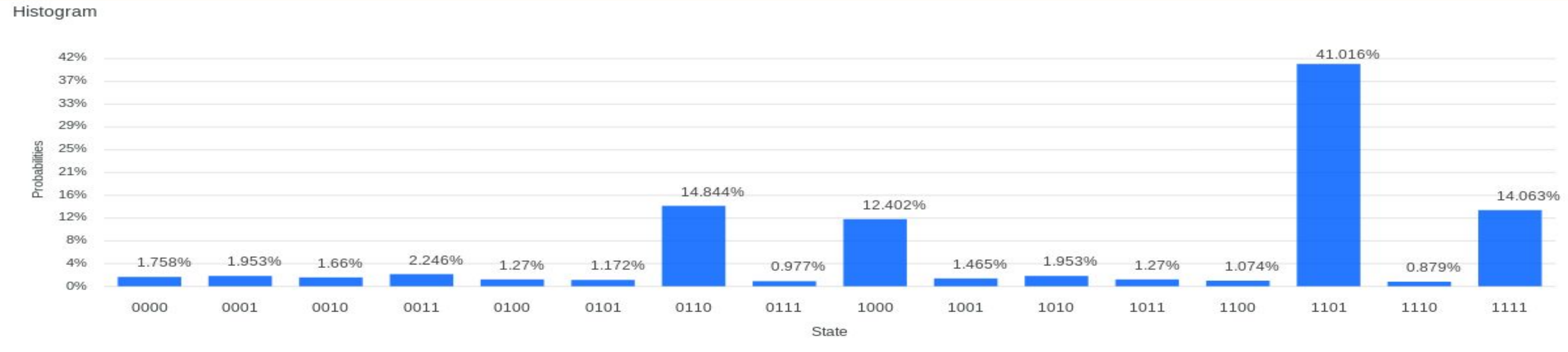


P
$|0\rangle^{\otimes n}$
Initialise
Oracle
Depends on T
A.A.
index
Grover iterations

Implementation

IBM-Q





*W: 1110, P: 10

Inbuilt Grover Algorithm

In [39]:
```
oracle_fn_0 = '''
c example DIMACS-CNF pattern 111000000 query:0
p cnf 4 3
1 2 3 4 0
1 2 3 -4 0
1 2 -3 4 0
'''
oracle_fn_1 = '''
c example DIMACS-CNF pattern 111000000 query:1
p cnf 4 6
1 2 -3 -4 0
1 -2 3 4 0
1 -2 3 -4 0
1 -2 -3 4 0
1 -2 -3 -4 0
-1 2 3 4 0
'''

oracle_0 = LogicalExpressionOracle(oracle_fn_0)
oracle_1 = LogicalExpressionOracle(oracle_fn_1)
```

In [41]:
```
backend = Aer.get_backend('qasm_simulator')
quantum_instance = QuantumInstance(backend, shots=1)

for i in range(M):
    ex = QuantumRegister(6)
    circuit += QuantumCircuit(ex)
    if(w[i]=='1'):
        grover = Grover(oracle=oracle_0,num_iterations=2)
#          grover.init_params(algo_input=circuit)
        circuit.append(grover.construct_circuit())
    else:
        grover = Grover(oracle=oracle_1,num_iterations=2)
        circuit.append(grover.construct_circuit())
circuit.measure(q[0:3],c[0:3])
job = execute(circuit, backend,shots=2048)
result = job.result()
print(result.get_counts(circuit))
```

Aritra Thesis Implementation

jupyter QPM-Copy1 Last Checkpoint: 16 hours ago (autosaved)

```python
    q = QuantumRegister(tot_q,'phi')
    c = ClassicalRegister(tot_q)
    circuit = QuantumCircuit(q,c)

    circ1(q,circuit,N,M,anc)

    bfa = ''.join('1' if w[i] == '0' else '0' for i in range(N))
    bfc = ''.join('1' if w[i] == '1' else '0' for i in range(N))
    bfg = ''.join('1' if w[i] == '2' else '0' for i in range(N))
    bft = ''.join('1' if w[i] == '3' else '0' for i in range(N))

    for r in range(0,int(sqrt(N-M+1))):
        ind = np.random.randint(0,M-1)
        if p[ind] == '0':
            circ2(q,circuit,bfa,ind,N,M,anc)
        elif p[ind] == '1':
            circ2(q,circuit,bfc,ind,N,M,anc)
#       elif p[ind] == '2':
#           c2 = circ2(q,bfg,ind,N,M,anc)
#       elif p[ind] == '3':
#           c2 = circ2(q,bft,ind,N,M,anc)

        circ3(q,circuit,N,M)

    circ4(q,c,circuit,N,M)

    simulator = Aer.get_backend('qasm_simulator')
    job = execute(circuit, simulator, shots=1024)
    result = job.result()
    counts = result.get_counts(circuit)
    for i in counts:
        print(i,counts[i])
```

References

- *Quantum Algorithms for pattern-matching in genomic sequences* by Aritra Sarkar
- *A Quantum Algorithm for Closest Pattern Matching* by P Mateus and Y Omar
- IBM Qiskit Documentation
    - https://qiskit.org/documentation/index.html
- Quantum Algorithm Implementations for Beginners
    - https://arxiv.org/pdf/1804.03719.pdf
- *Towards Data Science - Building Your Own Quantum Circuits in Python (With Colorful Diagrams)*
    - https://towardsdatascience.com/building-your-own-quantum-circuits-in-python-e9031b548fa7
- Grover Search Implementations
  https://community.qiskit.org/textbook/ch-algorithms/grover.html#implementation

Thank You