George Mason University | College of Science

Aspiring Scientists' Summer Internship Program

Comparisons of Classical and Quantum String Matching Algorithms

By MARGARET GAO, RACHEL HUANG, ARUL RHIK MAZUMDER, and Dr. Fei Li

String Matching Problem

- Given a text string **T**[1, n] and a pattern string **P**[1, m] identify any and all occurrences of the pattern in the text

- Could yield zero, one, or multiple such shifts

Example 1

T[1, 7] = “quantum” **P**[1, 3] = “ant”

Returns **s** = 2

Example 2

T[1, 6] = “banana” **P**[1, 3] = “ana”

Returns **s** = 1 and **s** = 3

Example 3

T[1, 3] = “are” **P**[1, 2] = “is”

Does not return anything

- A fundamental type of pattern matching problem relevant in text search, image processing, data compression, biological sequences and more

Noisy Intermediate-Scale Quantum (NISQ) Technology

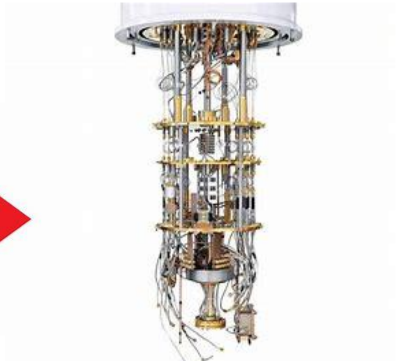
- Quantum Processing Units (QPU) can run algorithms for unsolvable classical problems.
- Units are currently limited by the number of available quantum bits (qubits)
- A lot of algorithms are not viable with current or near future NISQ technology
- Our work compare a novel quantum string-matching algorithm with classical variants

Novel Algorithm (Classical-Quantum Hybrid)

- The algorithm combines a classical sampling phase with a quantum search phase
- Avoids the qubit limitations of a pure quantum algorithm
- Incorporates many computational advantages of quantum algorithms
- Been used in quantum optimization, quantum search, and linear system solving

Text: “quantum”

Pattern: “ant”



GEORGE MASON UNIVERSITY | College of Science

Aspiring Scientists' Summer Internship Program

Novel Algorithm Explained

Input:

Text String ***T*** of length **n**

Pattern String ***P*** of length **m**

Example:

Text String: “quantum”

n = 7

Pattern String: “ant”

m = 3

Sampling:

Text randomly generates a set β substrings of α**m** length

Example:

β = 2

α = 4/3

Sampling = [“uant, “quan””]

Samplings are fed into quantum matching algorithm

Search:

Encoded into quantum registers.

Samples are bit-shifted and superimposed, then XORed to the pattern

Grover’s Search Algorithm searches for matches

Example:

Sampling [1] -> uant antu ntua tuan -> (uant XOR ant) -> 000

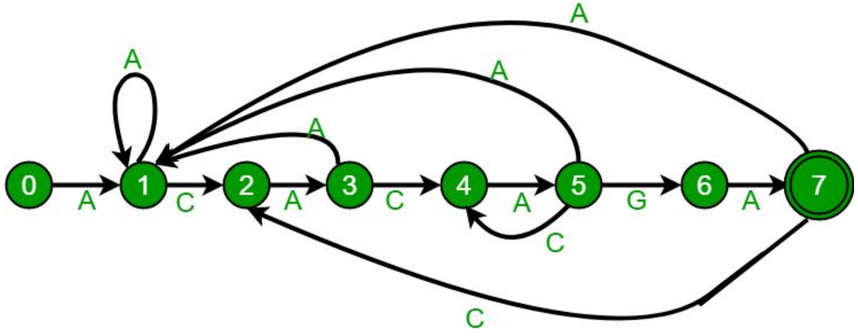
Grover’s Algorithm isolates the match state

Classic Algorithms Compared Against

- Rabin-Karp: Using a rolling hash to filter positions that match or don't match the pattern

1	a b a b a a c b a b	H("bab") = 293 H("bab") = 293	Yes	**Match**

- Finite Automata: Compares corresponding pattern and input characters. If the characters match, we progress to the right, if not, we go left back to the previous state. If the final state is reached, that means the pattern is found in the text.



Testing and Results

Table 3. Experimental results on long text, moderate pattern

algorithm	running time (maximum, minimum, average, ms)	missing matching
RK	1.00, 0.97, 1.00	0 out of 50
AHU	12.72, 3.84, 7.42	0 out of 50
our work	102.40, 26.59, 38.83	1 out of 50

Table 4. Experimental results on long text, short pattern, many occurrence

algorithm	running time (maximum, minimum, average, ms)	missing matching
RK	1.00, 1.00, 1.00	0 out of 50
AHU	3.96, 2.21, 2.97	0 out of 50
our work	32.86, 14.82, 17.11	0 out of 50

Table 5. Experimental results on long text, short pattern, moderate occurrence

algorithm	running time (maximum, minimum, average, ms)	missing matching
RK	0.997, 0.935, 0.983	0 out of 50
AHU	6.93, 1.88, 2.99	0 out of 50
our work	31.57, 14.92, 16.62	0 out of 50

Table 6. Experimental results on long text, short pattern, few occurrence

algorithm	running time (maximum, minimum, average, ms)	missing matching
RK	1.06, 0.996, 1.010	0 out of 50
AHU	12.72, 0.998, 3.84	0 out of 50
our work	32.01, 14.85, 18.56	0 out of 50

- Testing was done by matching DNA Patterns to Sequences from ENA Archive
- Classical Testing done using Intel Core i5 Processor
- Quantum Testing using the IBM QASM simulator on Qiskit

Conclusions and Future Work

- Quantum Algorithms run between 3-10 times worse than classical ones
- Grover’s search algorithm perform much slower in reality than in theory
- Quantum hardware needs to improve for quantum algorithms beat classical
- Optimizations of Grover’s Search Algorithm to bound errors
- Partial Database Searching superimposes the database before partitioning and searching

Thank You for Listening

Questions?