

```
PS C:\Users\geram> cd C:\Users\geram\OneDrive\Desktop\Quantum
PS C:\Users\geram\OneDrive\Desktop\Quantum> venv\Scripts\activate
(venv) PS C:\Users\geram\OneDrive\Desktop\Quantum> python qmazedyna1.py --visualize_grovers -
-seed 42 --cols 4 --rows 4 --loops 2

C:\Users\geram\OneDrive\Desktop\Quantum\qmazedyna1.py:3: DeprecationWarning: Using Qiskit
with Python 3.9 is deprecated as of the 2.1.0 release. Support for running Qiskit with Python 3.9 will
be removed in the 2.3.0 release, which coincides with when Python 3.9 goes end of life.

    from qiskit import QuantumCircuit, transpile

Using random seed: 42

--- Dynamic Quantum Maze Solver ---

Created 2 extra path(s) by removing walls.

Found 8512 potential paths. Using 14 qubits for a search space of 16384.

Found 8 valid solution(s).

Winning binary state(s): ['01010000010010', '01010001010111', '01010011010110',
'01010100011011', '10000010100110', '10000010101010', '10000010110010', '10000010110110']

Optimal number of Grover iterations: 36

--- Displaying Grover's Algorithm Step-by-Step Plots (Optimized) ---

WARNING: Visualizing 16384 states may be slow.

Will generate visualizations for steps: [ 0  4  8 12 16 20 24 28 32 36]

Running a single, consolidated simulation...

Simulation complete. Displaying plots from saved states...

Displaying plot for iteration 0. Please close the window to continue...

Displaying plot for iteration 4. Please close the window to continue...

Displaying plot for iteration 8. Please close the window to continue...

Displaying plot for iteration 12. Please close the window to continue...

Displaying plot for iteration 16. Please close the window to continue...

Displaying plot for iteration 20. Please close the window to continue...

Displaying plot for iteration 24. Please close the window to continue...

Displaying plot for iteration 28. Please close the window to continue...

Displaying plot for iteration 32. Please close the window to continue...

Displaying plot for iteration 36. Please close the window to continue...

--- Visualization Finished ---
```

--- Simulation Results ---

Counts: {'01010100011011': 123, '10000010100110': 125, '00000010000110': 1, '10000010110110': 119, '01010000010010': 121, '01010001010111': 150, '10000010101010': 111, '10000010110010': 139, '10110111001011': 1, '01010011010110': 132, '01001110011110': 1, '10100101111010': 1}

Quantum solver found solution: '01010001010111'

Solution is confirmed to be a valid path.

(venv) PS C:\Users\geram\OneDrive\Desktop\Quantum> python qmazedyna.py --visualize_grovers --seed 42 --cols 5 --rows 5 --loops 2

C:\Users\geram\OneDrive\Desktop\Quantum\qmazedyna.py:3: DeprecationWarning: Using Qiskit with Python 3.9 is deprecated as of the 2.1.0 release. Support for running Qiskit with Python 3.9 will be removed in the 2.3.0 release, which coincides with when Python 3.9 goes end of life.

```
from qiskit import QuantumCircuit, transpile
```

Using random seed: 42

--- 1. Generating Maze ---

Created 2 extra path(s) by removing walls.

Maze Layout:

S W . . .

. W . . .

. . . W .

. W . W .

. . . W E

--- 2. Converting Maze to Graph ---

Graph created with 19 nodes and 21 edges.

Start Node ID: 0 @ (0, 0)

End Node ID: 24 @ (4, 4)

--- 3. Finding Classical Solution Path(s) ---

Found 8 classical path(s).

--- 4. Running Quantum Search (Simulation) ---

Found 8 valid solution(s).

Creating a search space of N=8 (using 3 qubits).

Winning binary state(s): ['110', '100', '001', '010', '000', '011', '101', '111']

All states are solutions. No search needed.

Search Space N=8, Solutions M=8

Optimal Iterations: 0

--- Simulation Results ---

Counts: {'111': 147, '110': 137, '101': 120, '001': 125, '100': 131, '011': 126, '010': 125, '000': 113}

Quantum solver found solution: '111'

Solution is confirmed to be a valid path.

[Grid visualization saved to maze_solution_grid.png]

[Histogram saved to quantum_search_histogram.png]

--- Script Finished ---

(venv) PS C:\Users\geram\OneDrive\Desktop\Quantum> python final.py --loops 2 --rows 5 --cols 5 --seed 42

C:\Users\geram\OneDrive\Desktop\Quantum\final.py:3: DeprecationWarning: Using Qiskit with Python 3.9 is deprecated as of the 2.1.0 release. Support for running Qiskit with Python 3.9 will be removed in the 2.3.0 release, which coincides with when Python 3.9 goes end of life.

```
from qiskit import QuantumCircuit, transpile
```

Using random seed: 42

[Maze Gen] Created 2 extra path(s) by removing walls.

--- Running Quantum Solver ---

--- Running Classical (A*) Solver ---

--- FINAL RESULTS ---

Maze Dimensions: 5x5

Classical (A*) Solver:

- Time Taken: 0.1293 ms

- Path Found: Yes
- Nodes Explored: 16
- Path Length: 8

Quantum (Grover's) Solver:

- Total Time: 352.0971 ms
 - Classical Prep Time: 85.7174 ms
 - Quantum Sim Time: 266.3685 ms
- Search Space Size (N): 16384 (encoded in 14 qubits)
- Number of Solutions (M): 8
- Oracle Queries (Iterations): 36 ($\sqrt{N/M}$)
- Success Probability: 14.65%
- Path Found: Yes
- Path Length: 16

(venv) PS C:\Users\geram\OneDrive\Desktop\Quantum>