Comparison of Heuristic Measures and Search Algorithms on Maze Problems

Muhammad Jamil Sani
GitHub Repository

February 13, 2025

1 Introduction

Maze-solving algorithms play a significant role in pathfinding and artificial intelligence applications. In this report, we explore and compare the performance of A* and Greedy Best-First Search (GBFS) algorithms using two heuristic measures: Manhattan Distance and Octile Distance. The evaluation is conducted on three different mazes of varying sizes.

2 Heuristic Measures

We consider the following heuristic measures for our comparison:

- **Manhattan Distance**: This heuristic measures the sum of absolute differences between the current position and the goal. It is best suited for grid-based pathfinding where only horizontal and vertical movements are allowed. While simple and efficient, it does not account for diagonal movement, which may lead to suboptimal path exploration in some cases.
- **Octile Distance**: This heuristic is a variation of the Manhattan Distance that accounts for diagonal movement. It is computed as the sum of horizontal and vertical differences, with diagonal steps weighted accordingly. This heuristic provides a more accurate estimate in scenarios where diagonal movement is permitted, leading to potentially more efficient paths.

3 Implementation Details

The algorithms were implemented in Python using a priority queue for efficient frontier management. Three maze sizes (30x30, 35x35, and 40x40) were tested. Each heuristic was applied to both A* and GBFS to compare efficiency. The number of nodes explored, execution time, and path cost were measured for each configuration.

4 Experimental Results

The following results summarize the number of nodes explored, execution time, and path cost across different algorithms and heuristics.

Maze Size	Algorithm	Nodes Explored	Execution Time (s)	Path Cost
30x30	A* (Manhattan)	275	0.001832	76
30x30	A* (Octile)	292	0.001924	76
30x30	GBFS (Manhattan)	114	0.000942	76
30x30	GBFS (Octile)	174	0.001110	76
35x35	A* (Manhattan)	241	0.001332	96
35x35	A* (Octile)	246	0.001501	96
35x35	GBFS (Manhattan)	163	0.001252	96
35x35	GBFS (Octile)	153	0.00138	96
40x40	A* (Manhattan)	407	0.002349	116
40x40	A* (Octile)	475	0.002921	116
40x40	GBFS (Manhattan)	310	0.002418	116
40x40	GBFS (Octile)	219	0.001815	116

Table 1: Comparison of nodes explored, execution time and path cost across different the algorithms and heuristics (The result above is not deterministic as it varies across different runtime/execution).

4.1 Nodes Explored

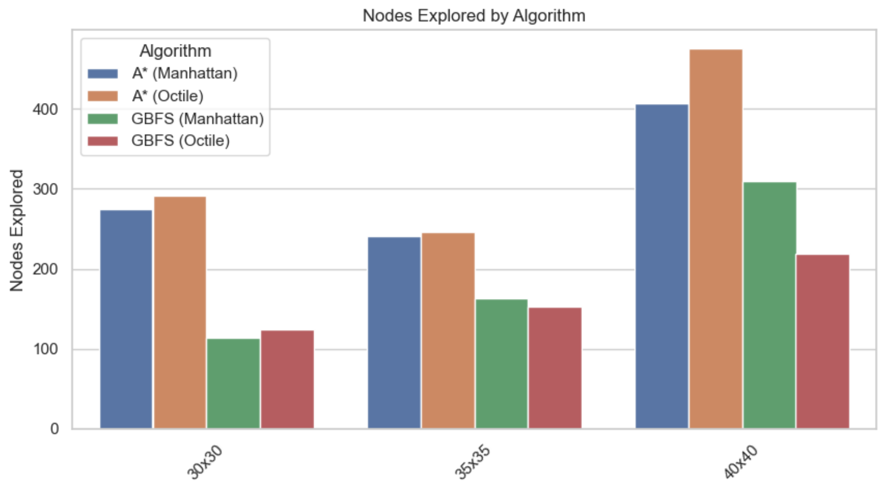

Figure 1: Comparison Nodes Explored for different heuristics.

4.2 Execution Time

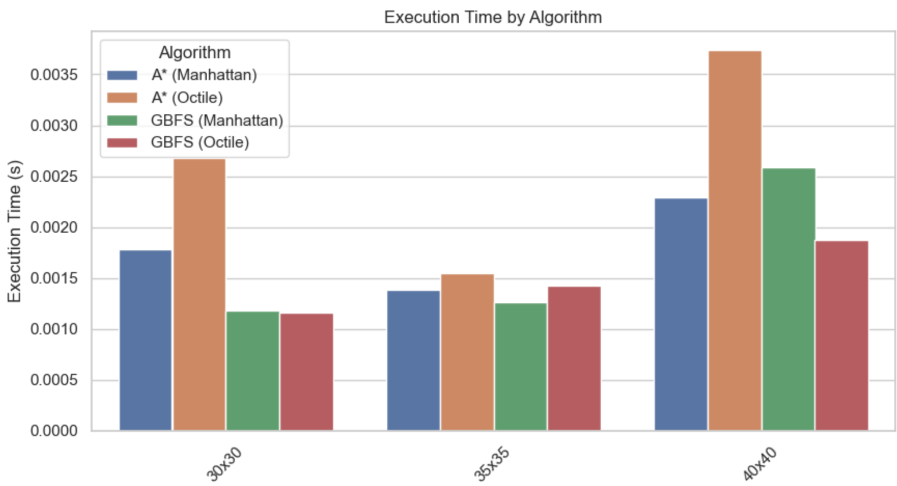

Figure 2: Execution time comparison for different heuristics.

4.3 Path Cost

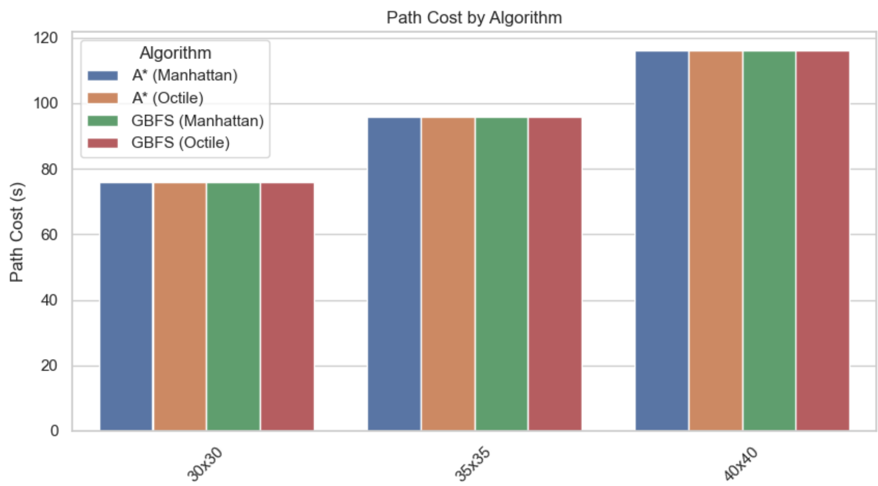

Figure 3: Comparison of path costs for different algorithms.

5 Analysis and Discussion

The results indicate that A* consistently finds the shortest paths but requires exploring more nodes, making it computationally expensive. GBFS explores fewer nodes but does not always yield the optimal solution.

Regarding heuristics, Manhattan Distance performs well in environments where movement is strictly horizontal or vertical. However, it may lead to unnecessary expansions when diagonal movement is beneficial. The Octile Distance heuristic provides a more accurate estimate by considering diagonal movement, reducing node expansions in such scenarios and improving computational efficiency.

The choice of heuristic significantly affects search performance. The Manhattan heuristic can sometimes overestimate the cost due to its strict grid-based approach, leading to longer search times. Octile Distance improves performance in scenarios where diagonal movements are beneficial, making it a more flexible choice for varied maze layouts.

6 Conclusion

In this study, we evaluated the performance of A* and GBFS using Manhattan and Octile Distance heuristics. A* provided optimal paths at the cost of higher computational overhead, while GBFS was faster but less optimal.

The Manhattan heuristic is effective in purely grid-based movement, whereas Octile Distance offers better efficiency when diagonal movements are allowed. Future work could explore additional heuristics or hybrid approaches for further optimization.

7 Appendix

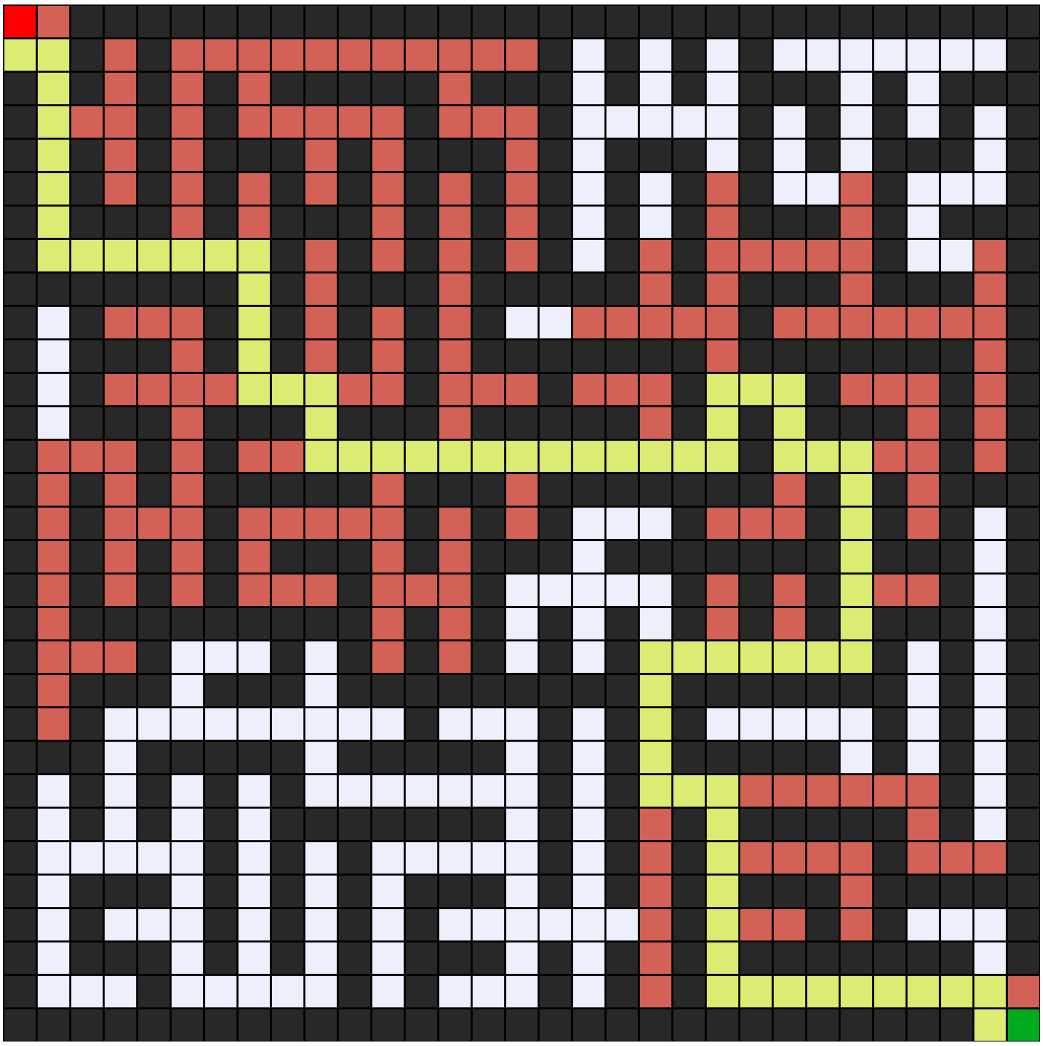

Figure 4: A* Manhattan Distance nodes explored for the 30x30 maze

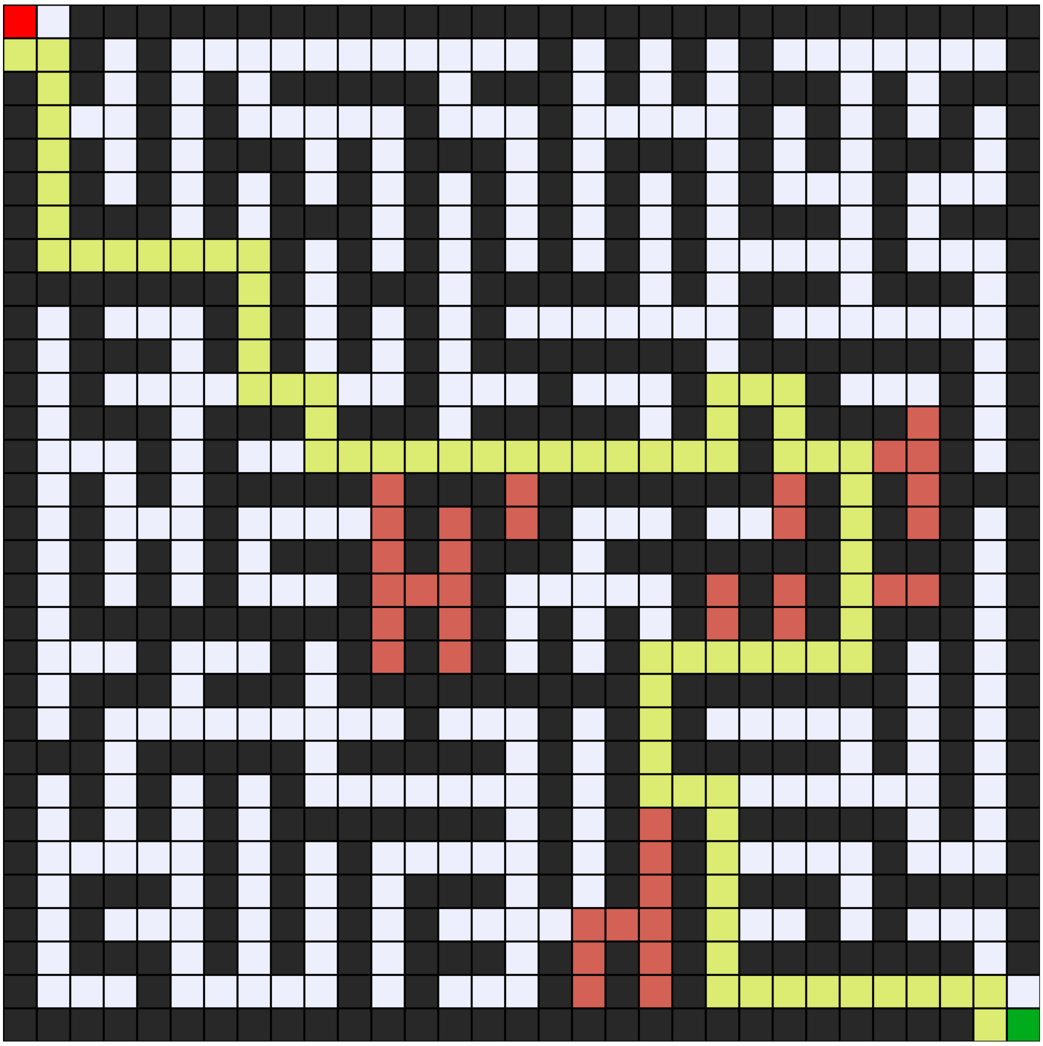

Figure 5: GBFS Manhattan Distance nodes explored for the 30x30 maze

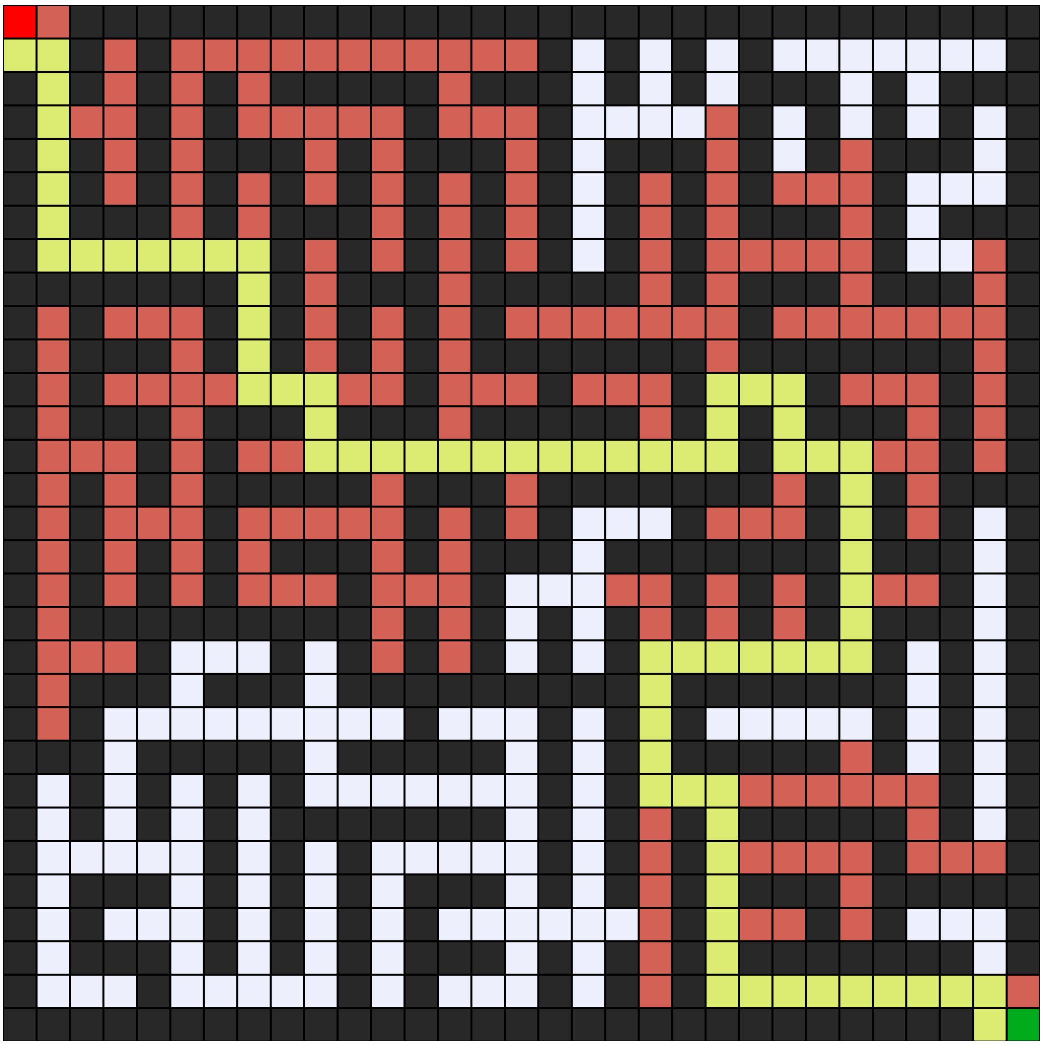

Figure 6: A* Octile Distance nodes explored for the 30x30 maze

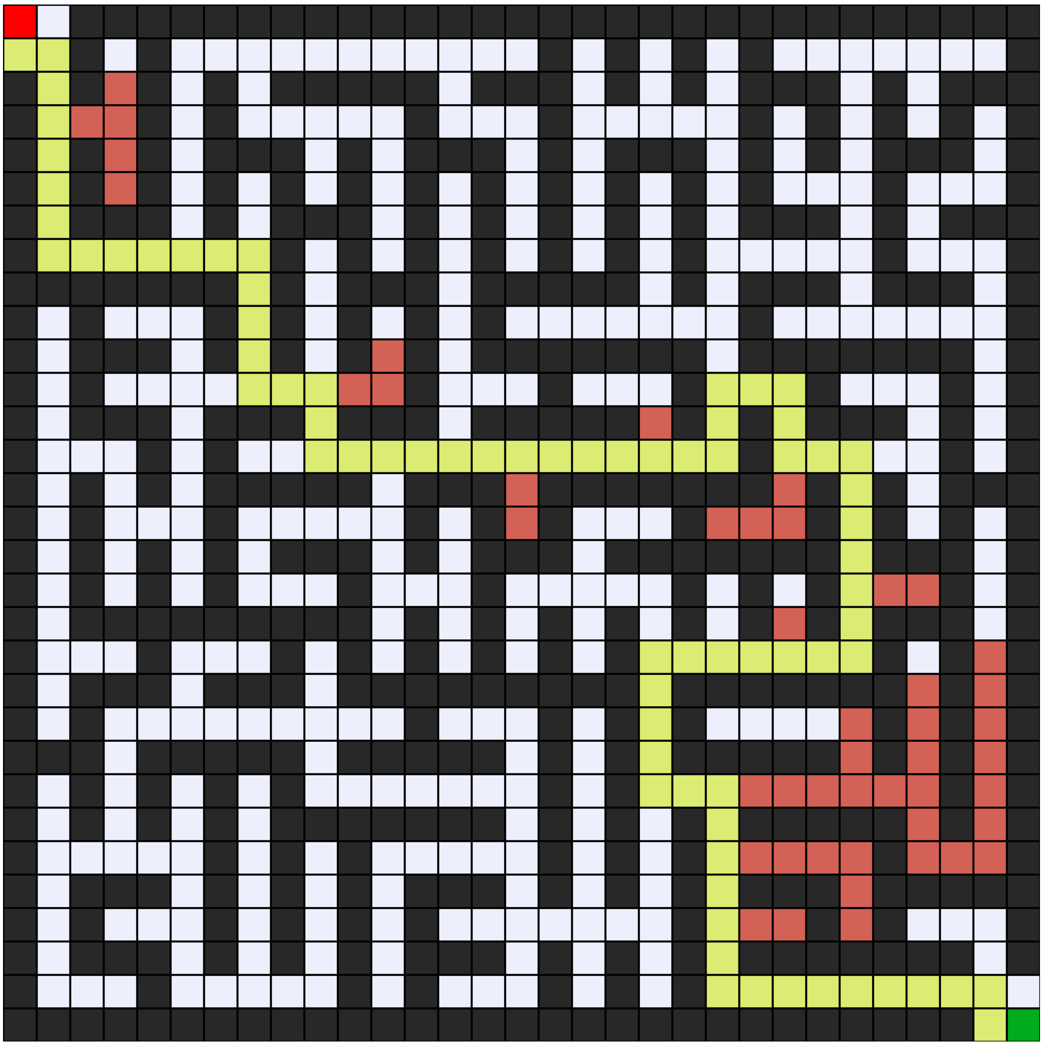

Figure 7: GBFS Octile Distance nodes explored for the 30x30 maze

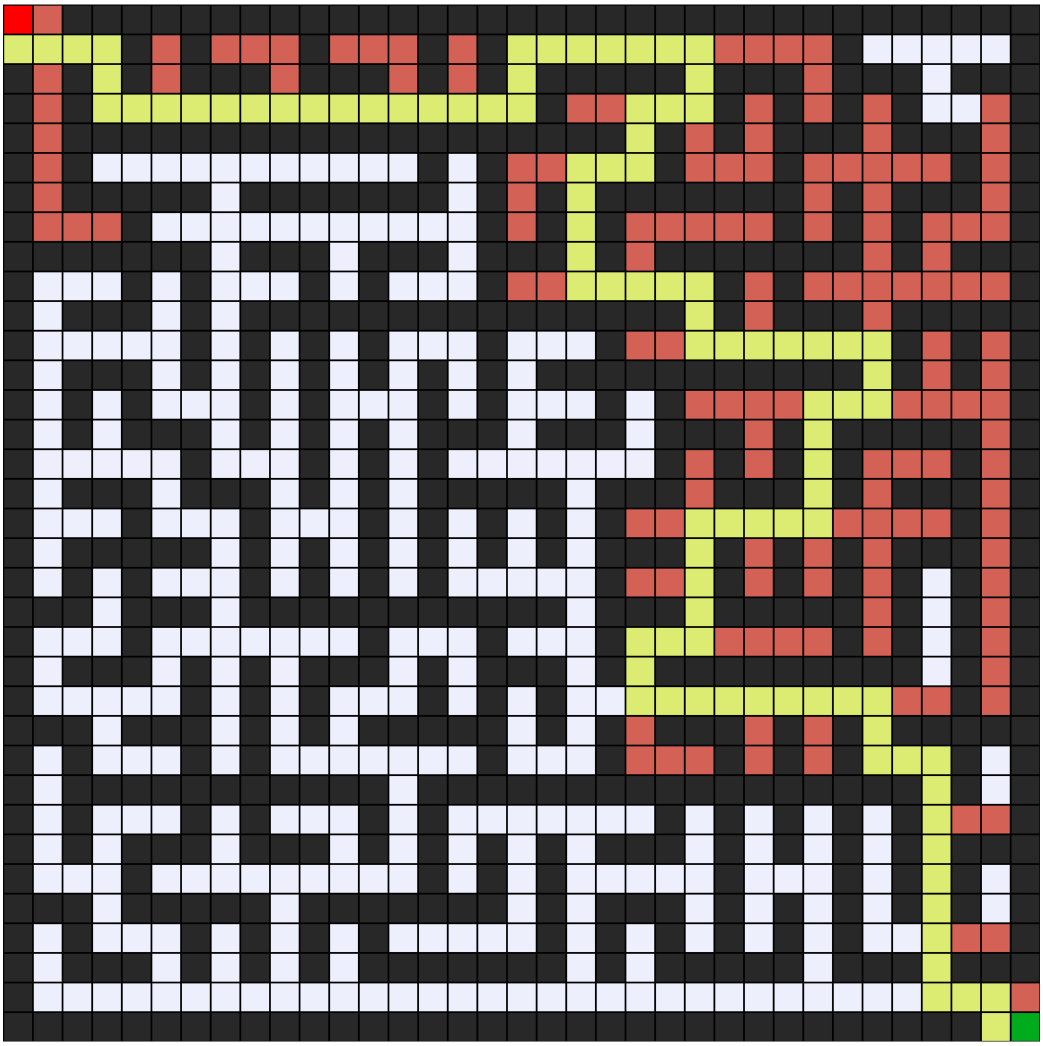

Figure 8: A* Manhattan Distance nodes explored for the 35x35 maze

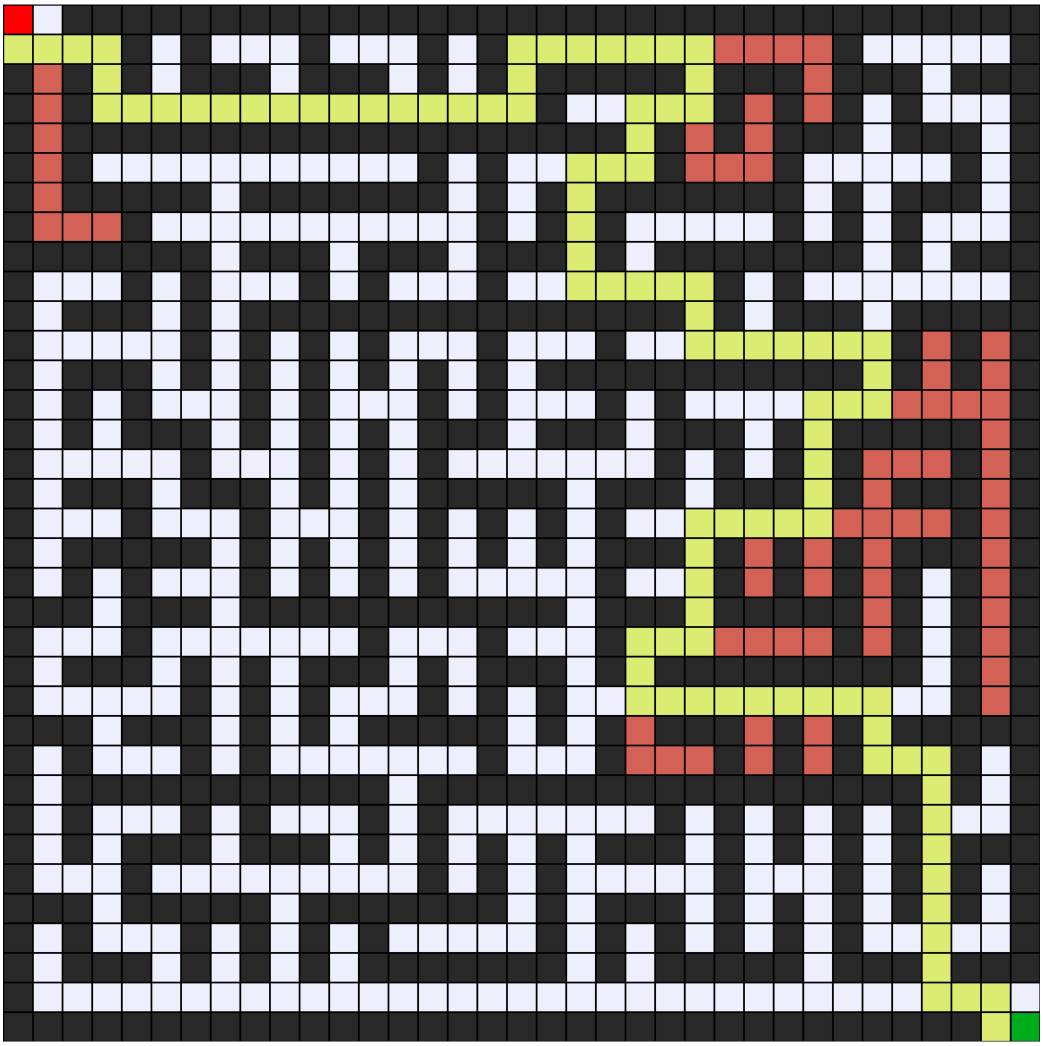

Figure 9: GBFS Manhattan Distance nodes explored for the 35x35 maze

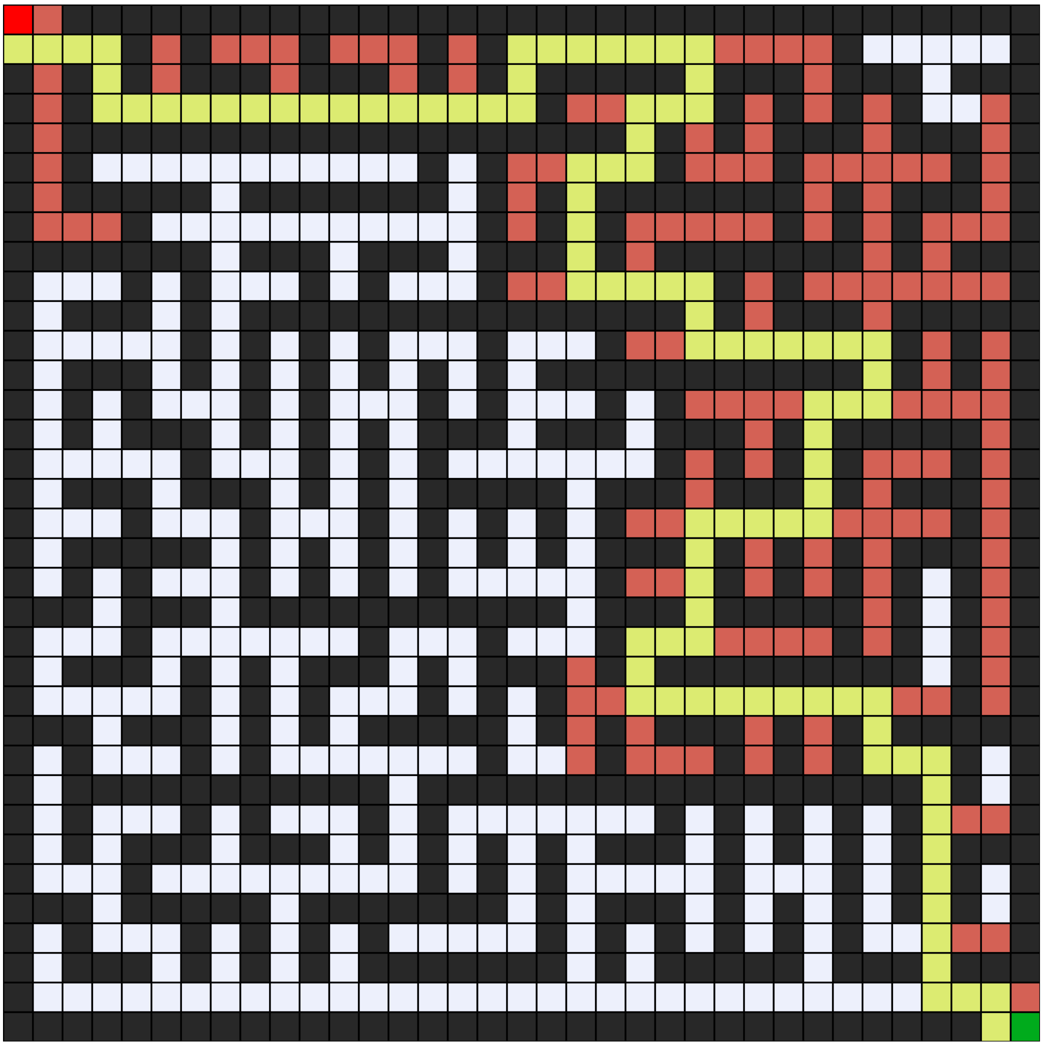

Figure 10: A* Octile Distance nodes explored for the 35x35 maze

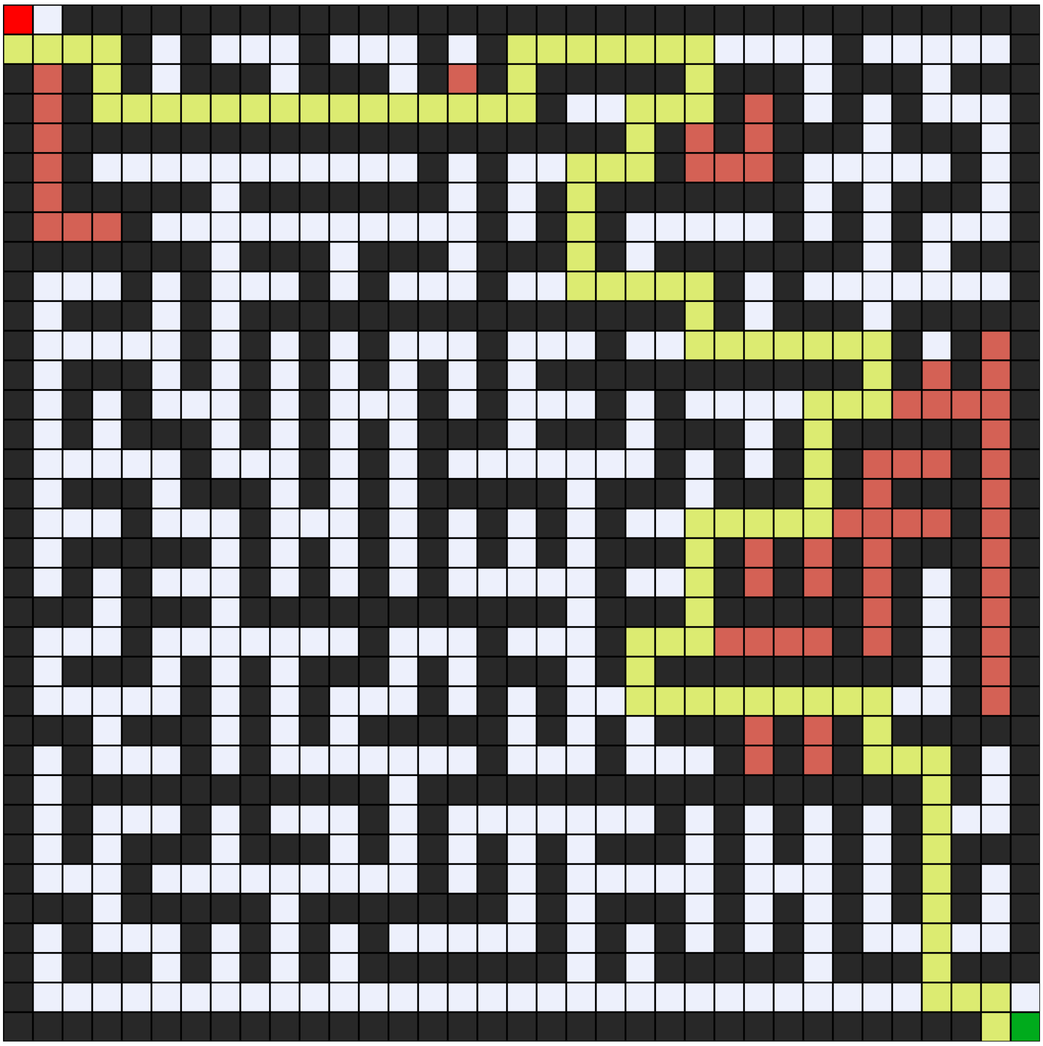

Figure 11: GBFS Octile Distance nodes explored for the 35x35 maze

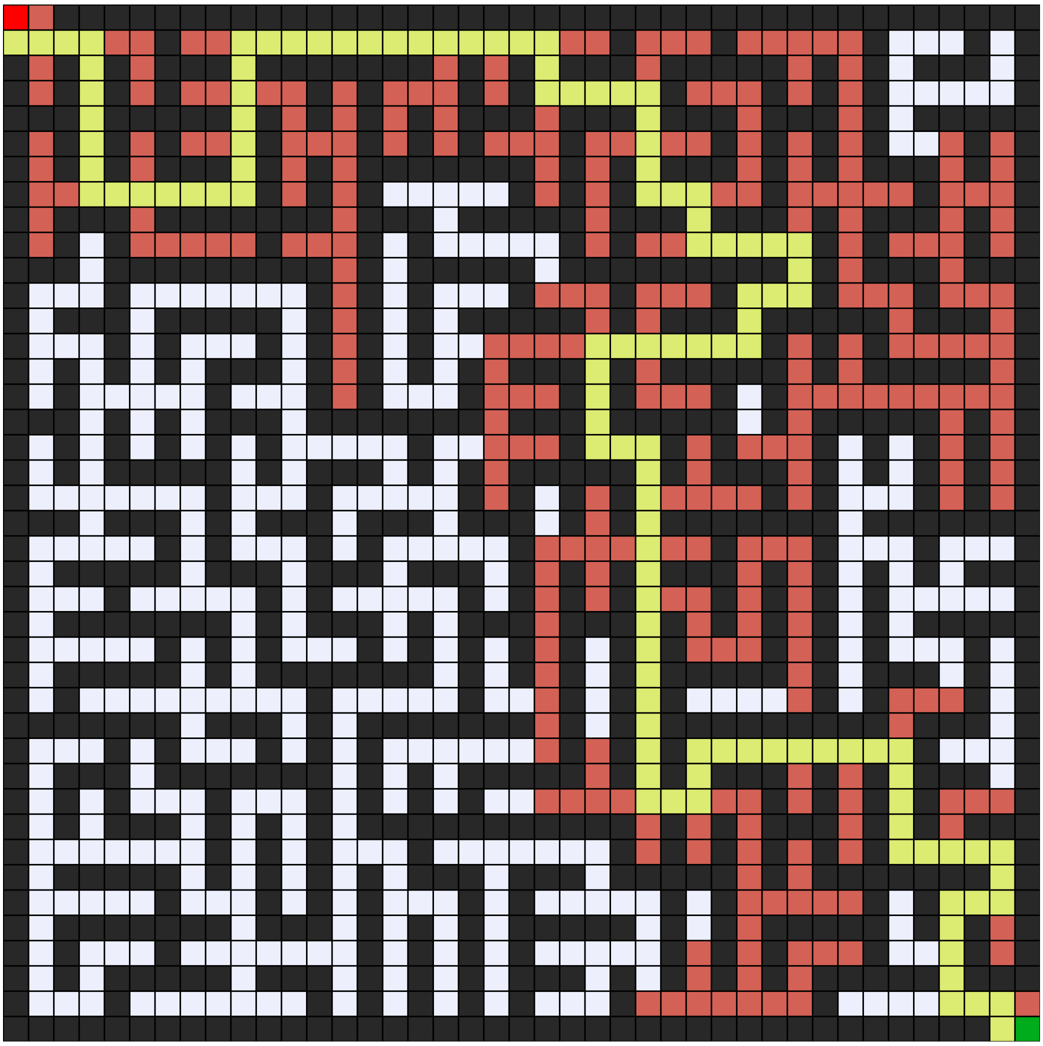

Figure 12: A* Manhattan Distance nodes explored for the 40x40 maze

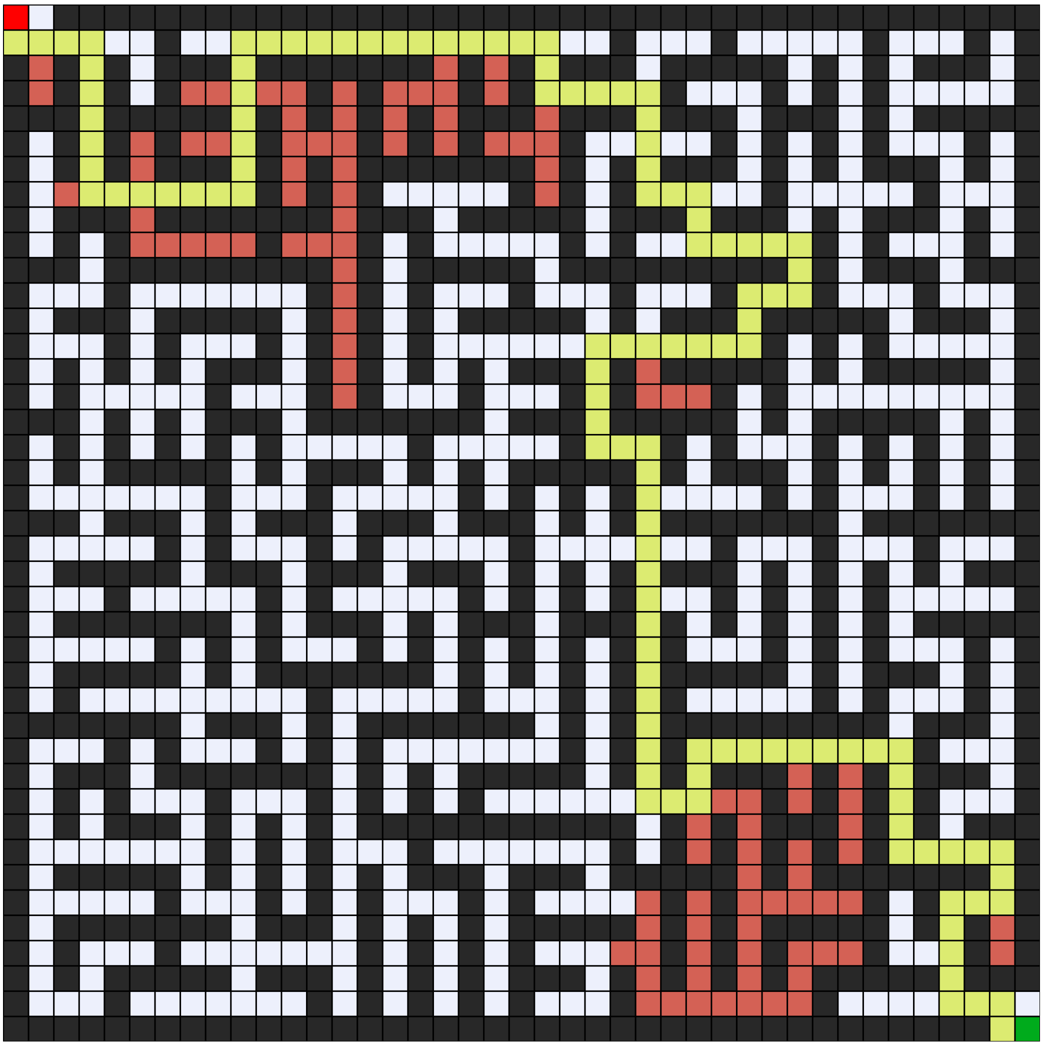

Figure 13: GBFS Manhattan Distance nodes explored for the 40x40 maze

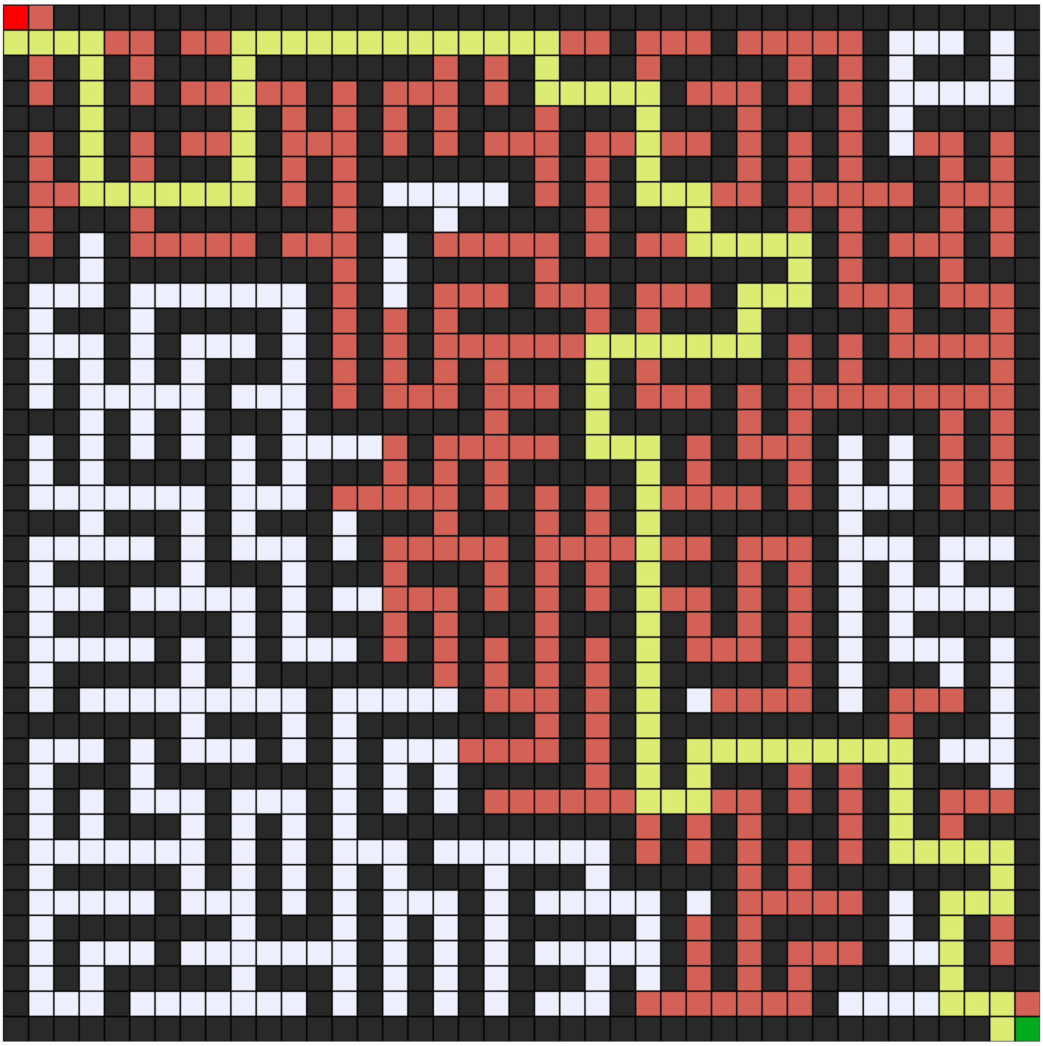

Figure 14: A* Octile Distance nodes explored for the 40x40 maze

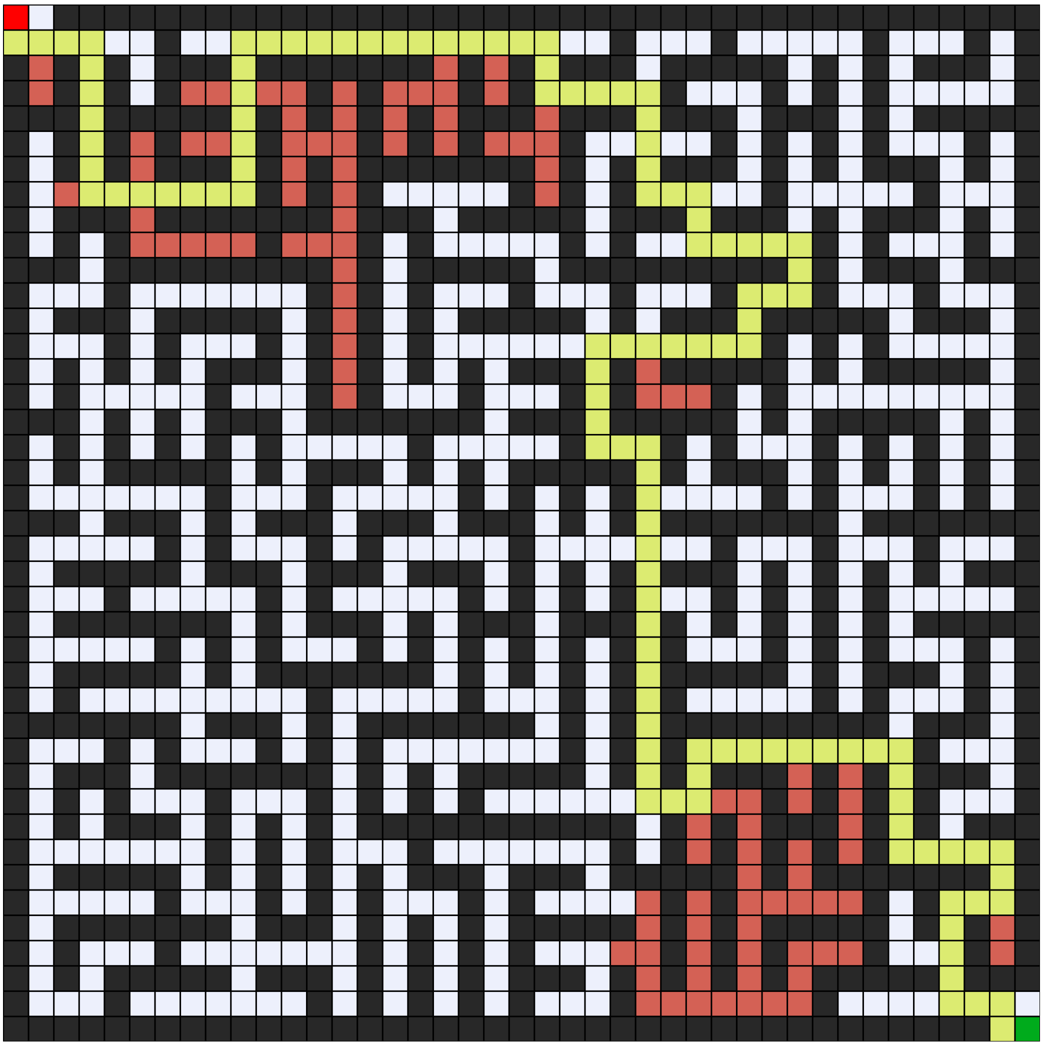

Figure 15: GBFS Octile Distance nodes explored for the 40x40 maze