ME5413-Autonomous Mobile Robotics

Homework_3_Localisation

Name: Ma Xurong

Matric Number: A0303475H

NusNetID: E1372563

2025/03/25

Task 1: Global Planning

The objective of Task 1 is to plan a path for every pair of the five key locations on the Level 2 map of VivoCity. For each start and end point pair, the path should be visualized and the total travel distance calculated. The A* algorithm is to be used to accomplish this task. Additionally, one to two improvement approaches should be explored and implemented to enhance the performance or results of the algorithm.

Firstly, since the human body occupies a circular area with a radius of no less than 0.3m, it is necessary to perform obstacle dilation on the map. That is, to expand the boundaries of the obstacles within a certain range, so as to ensure that the planned path has sufficient safety distance and avoid collisions with obstacles. This can be achieved on a binary map through the $binary\_dilation$ morphological dilation function in the $SciPy$ library. As shown in Figure 1, the left image is the original binary map before dilation, and the right image is the map after dilation.

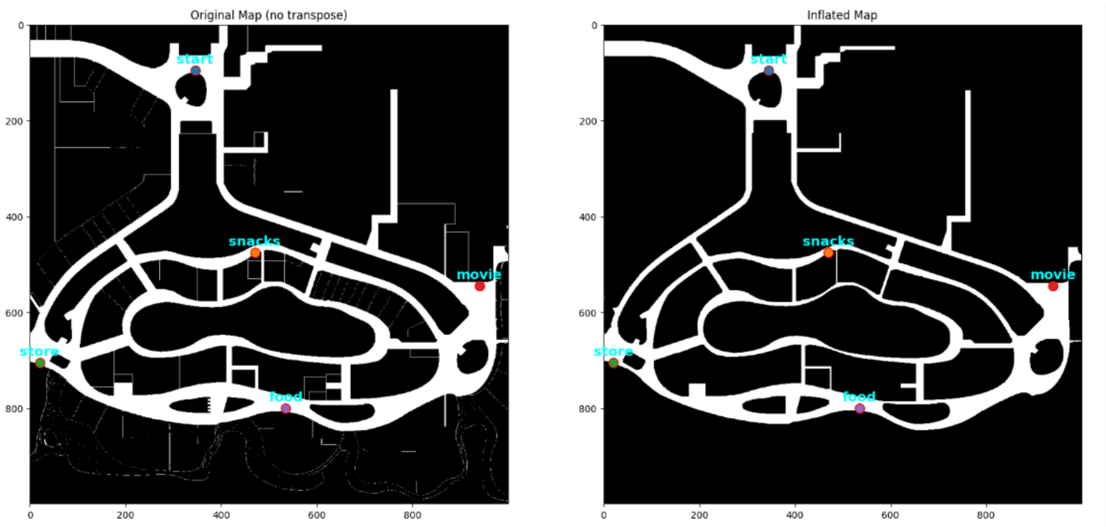


Figure 1. Binary maps before and after expansion.

After completing obstacle inflation on the map, path searching can be performed using the A* algorithm. First, the Euclidean distance is defined as the heuristic function to estimate the shortest distance from the current position to the target point. This is combined with the actual cost of the path taken so far (g value) to calculate the total cost ($f = g + h$). The algorithm considers 8 possible movement directions (including straight and diagonal moves), and different movement costs are assigned: $0.2m$ for straight directions and $0.282m$ for diagonal ones. During the search, Python's $heapq$ is used to build a priority queue for managing the open set, always selecting the node with the smallest total cost for expansion. A dictionary $cost\_so\_far$ is used to record the minimum cost to reach each node, and $came\_from$ is used to track the path origin, allowing the full path to be reconstructed by backtracking once the target is reached. The path length is then converted into actual physical distance using the map resolution MAP_RES, and the algorithm's runtime is also recorded for performance evaluation.

Finally, the output data and path visualization are displayed. As shown in Figure 2, which presents the results for the path from the start point to the snacks point, the total distance is $28.99m$, the number of visited cells is 4074, and the computation time is 0.3621 seconds. The data and visualizations for the other points will be included in the appendix.

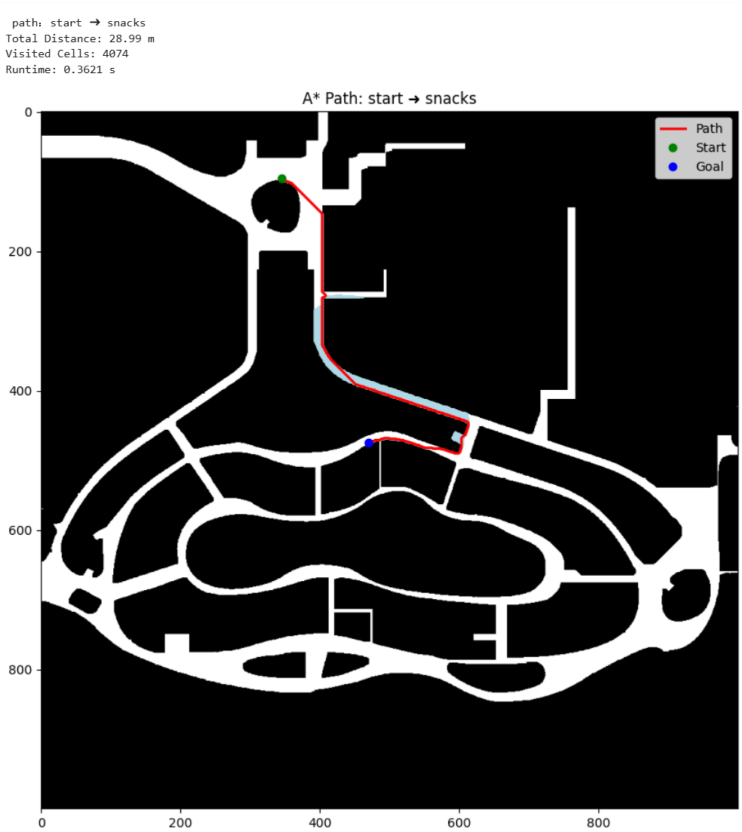


Figure 2. Binary maps before and after expansion.

Table 1. The distances between each pair of locations.

To\ From	start	snacks	store	movie	food
Start	0	28.99	32.04	36.14	46.72
snacks	28.99	0	24.80	25.07	27.14
Store	32.04	24.80	0	51.42	23.04
Movie	36.14	25.07	51.42	0	40.37
Food	46.72	27.14	23.04	40.37	0

Next, the A* algorithm is simplified to implement Greedy Best-First Search, and the basic RRT (Rapidly-exploring Random Tree) algorithm is also implemented. The results of these two methods are then compared. As shown in Figure 3, subfigures (a) and (b) display the resulting paths generated by Greedy Best-First Search and the basic RRT algorithm, respectively.

(a) (b)

Figure 3. Binary maps before and after expansion.

A detailed comparison of the path length, computation time, and number of visited nodes is presented in Tables in the appendix. As shown in the tables, in terms of the final path length, the *A* algorithm* achieves the shortest average distance at 34.55 meters, demonstrating good optimality. The Greedy Best-First Search produces similar results, with slightly longer average paths. The RRT method, however, results in the longest average distance of 213.81 meters, indicating weaker path convergence and a tendency to expand into unnecessary regions.

In terms of computation time, Greedy Best-First Search is the fastest, with an average runtime of just 0.055 seconds, thanks to its simple heuristic-driven strategy. RRT, on the other hand, has the longest runtime, averaging 12.3359 seconds, mainly due to its sampling-based approach, which often generates many redundant nodes in complex environments.

Regarding the number of visited grid cells, RRT visits only 479.8 cells on average, whereas A* and Greedy Best-First Search each explore over 2,000 cells on average. While this may seem counterintuitive, the results indicate that although RRT appears to visit fewer grid cells, its random sampling and tree structure do not effectively guide the search toward the optimal path. As a result, the paths generated by RRT are more curved and unnecessarily long.

Task 2: The "Travelling Shopper" Problem

This problem can be formulated as the classical Travelling Salesman Problem (TSP): starting from a given point, the goal is to visit all specified locations exactly once and return to the starting point, while minimizing the total travel distance. As an NP-hard problem, it is well-suited for combinatorial optimization approaches.

Based on the pairwise distance matrix of the five key locations constructed in Task 1, this task is addressed using the following two methods: The first one is Brute-force Enumeration. All possible visiting orders are exhaustively generated (i.e., all permutations of the 4 intermediate locations, resulting in 4! = 24 combinations). For each route, the total travel distance (including the return to the starting point) is calculated, and the shortest path is selected. This method guarantees the optimal solution and is suitable when the number of locations is small. In our implementation, we use $itertools.permutations()$ to generate all possible orders of the four destinations and compute the total distance for each route to find the optimal one.

The second one is Nearest Neighbor Heuristic. Starting from the initial location, the algorithm repeatedly visits the nearest unvisited location until all points are visited, and finally returns to the starting point. Although it does not guarantee the optimal solution, it is computationally efficient and suitable for quick planning in practical applications. In this implementation, at each step, the nearest unvisited point is selected until the full tour is completed.

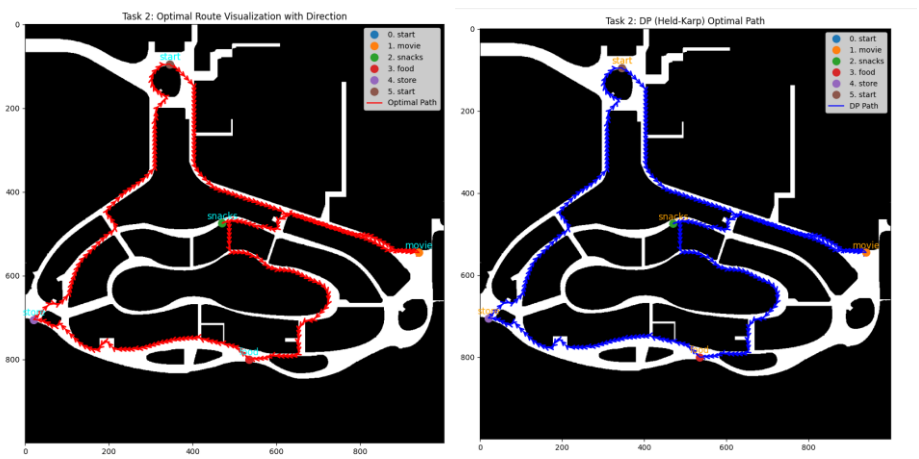


Figure 4. The experimental results across different algorithms.

Table 2. The comparison of experimental results across different algorithms.

Method	Distance	Time
Brute-force Enumeration	143.43m	0.0031s
Nearest Neighbor Heuristic	143.43m	0.0028s

The results obtained from both experimental methods are identical — including the visiting sequence, the total distance, and the segment-by-segment distances. However, the computation time differs slightly: the Brute-force Enumeration method took 0.0031 seconds, while the Nearest Neighbor Heuristic completed in 0.0028 seconds. This indicates that for the current small-scale problem (with only 4 intermediate points), both the Brute-force Enumeration method and the Nearest Neighbor Heuristic are capable of successfully finding the global optimal solution. Although the difference in computation time is relatively small, the heuristic algorithm still demonstrates an advantage in computational efficiency. Especially for medium- to large-scale problems, the complexity of the brute-force method grows factorially, whereas the nearest neighbor algorithm can provide a feasible solution in a much shorter time, making it more suitable for the demands of rapid path planning in practical applications.

Task 3: Github upload

https://github.com/Ma-Romy/planning-project.git

Appendix

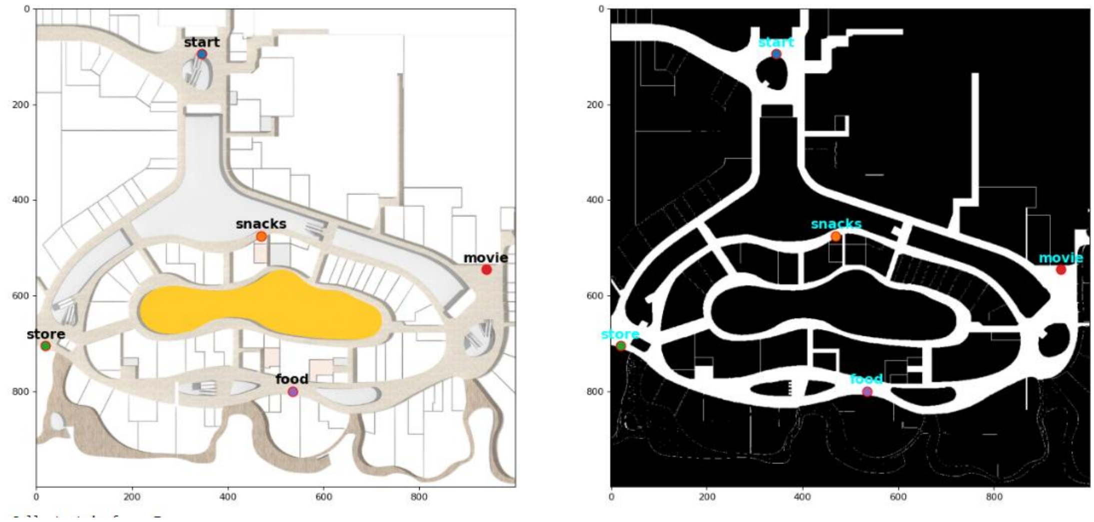


Figure 1. Map images before and after binarization processing

Path Distance Comparison (A* vs Greedy vs RRT)

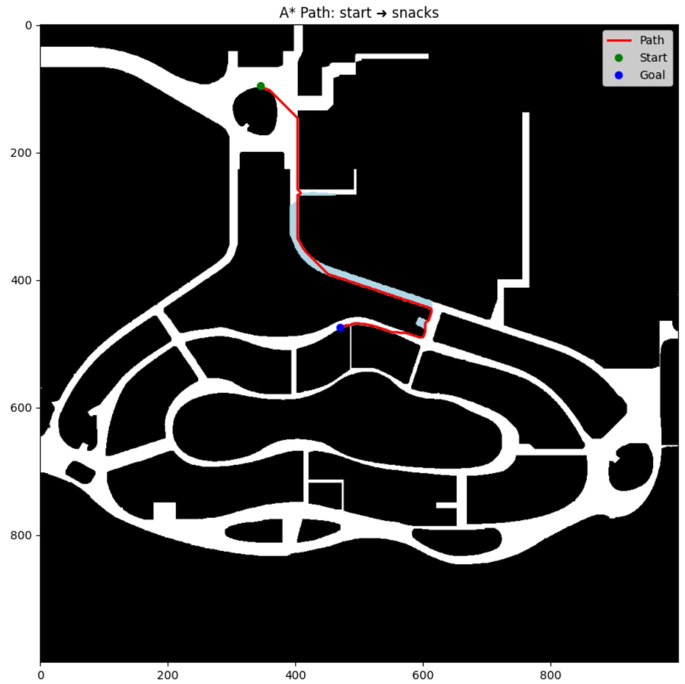
	From	To	A* (m)	Greedy (m)	RRT (m)	Greedy Δ (m)	RRT Δ (m)	Greedy Error (%)	RRT Error (%)
1	start	store	32.04	32.15	384.69	0.11	352.65	0.34	1100.66
7	store	start	32.04	32.15	384.69	0.11	352.65	0.34	1100.66
15	food	snacks	27.14	34.35	229.14	7.21	202.00	26.57	744.29
6	snacks	food	27.14	34.35	229.14	7.21	202.00	26.57	744.29
14	food	start	46.72	47.97	300.60	1.25	253.88	2.68	543.41
3	start	food	46.72	47.97	300.60	1.25	253.88	2.68	543.41
9	store	food	23.04	23.81	138.77	0.77	115.73	3.34	502.30
16	food	store	23.04	23.81	138.77	0.77	115.73	3.34	502.30
11	movie	snacks	25.07	28.89	146.21	3.82	121.14	15.24	483.21
5	snacks	movie	25.07	28.89	146.21	3.82	121.14	15.24	483.21
2	start	movie	36.14	36.48	191.50	0.34	155.36	0.94	429.88
10	movie	start	36.14	36.48	191.50	0.34	155.36	0.94	429.88
0	start	snacks	28.99	29.25	150.34	0.26	121.35	0.90	418.59
4	snacks	start	28.99	29.25	150.34	0.26	121.35	0.90	418.59
8	store	movie	51.42	51.41	251.77	-0.01	200.35	-0.02	389.63
12	movie	store	51.42	51.41	251.77	-0.01	200.35	-0.02	389.63
13	movie	food	40.37	41.61	131.27	1.24	90.90	3.07	225.17
17	food	movie	40.37	41.61	131.27	1.24	90.90	3.07	225.17

```
 Average Path Lengths
- A*     Avg: 34.55 m
- Greedy Avg: 36.21 m
- RRT    Avg: 213.81 m

 Average Distance Difference & Error (relative to A*)
- Greedy Δ: 1.67 m | Error: 5.90 %
- RRT    Δ: 179.26 m | Error: 537.46 %
```

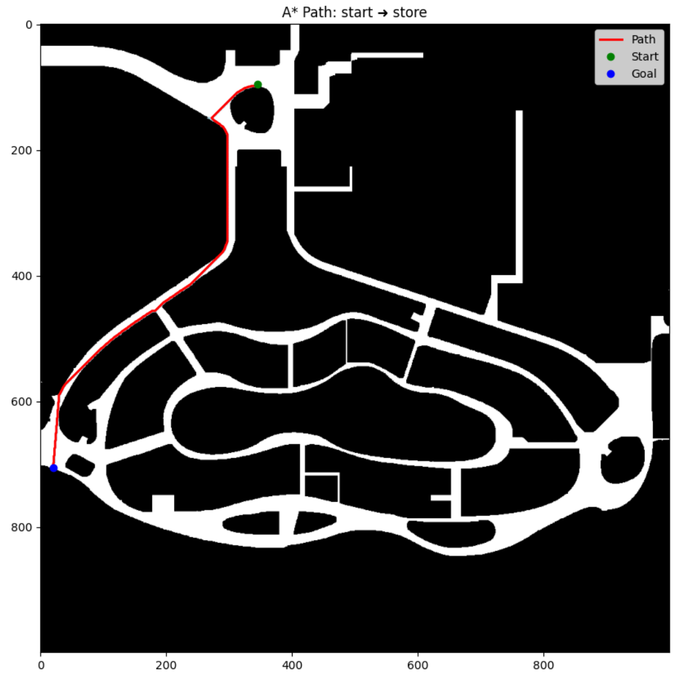
	From	To	A* Time (s)	Greedy Time (s)	RRT Time (s)	A* Visited	Greedy Visited	RRT Visited	Time Ratio (Greedy/A*)	Time Ratio (RRT/A*)	Visited Ratio (Greedy/A*)	Visited Ratio (RRT/A*)
6	store	food	0.0189	0.0218	9.2592	541	536	580	1.15	489.78	0.99	1.07
7	food	store	0.0189	0.0218	9.2592	541	536	580	1.15	489.78	0.99	1.07
1	store	start	0.0288	0.0264	13.8701	695	677	590	0.92	480.87	0.97	0.85
0	start	store	0.0288	0.0264	13.8701	695	677	590	0.92	480.87	0.97	0.85
12	start	snacks	0.2613	0.0848	55.6964	4074	3815	1191	0.32	213.13	0.94	0.29
13	snacks	start	0.2613	0.0848	55.6964	4074	3815	1191	0.32	213.13	0.94	0.29
10	start	movie	0.1535	0.0391	19.0780	1659	1651	728	0.25	124.27	1.00	0.44
11	movie	start	0.1535	0.0391	19.0780	1659	1651	728	0.25	124.27	1.00	0.44
15	movie	store	0.0900	0.0645	1.8919	1903	1364	286	0.72	21.01	0.72	0.15
14	store	movie	0.0900	0.0645	1.8919	1903	1364	286	0.72	21.01	0.72	0.15
4	food	start	0.4371	0.0739	7.8283	4302	3460	425	0.17	17.91	0.80	0.10
5	start	food	0.4371	0.0739	7.8283	4302	3460	425	0.17	17.91	0.80	0.10
9	snacks	movie	0.1046	0.0178	0.6110	1335	843	118	0.17	5.84	0.63	0.09
8	movie	snacks	0.1046	0.0178	0.6110	1335	843	118	0.17	5.84	0.63	0.09
3	snacks	food	0.4543	0.0794	2.4363	4708	3096	282	0.17	5.36	0.66	0.06
2	food	snacks	0.4543	0.0794	2.4363	4708	3096	282	0.17	5.36	0.66	0.06
16	movie	food	0.4775	0.0872	0.3515	5509	2686	118	0.18	0.74	0.49	0.02
17	food	movie	0.4775	0.0872	0.3515	5509	2686	118	0.18	0.74	0.49	0.02

```
Task 2 Optimal Route (Brute Force TSP):
Visiting Order : start → movie → snacks → food → store → start
Total Distance : 143.43 m

Segment Distances (in meters):
start → movie : 36.14 m
movie → snacks : 25.07 m
snacks → food : 27.14 m
food → store : 23.04 m
store → start : 32.04 m
```

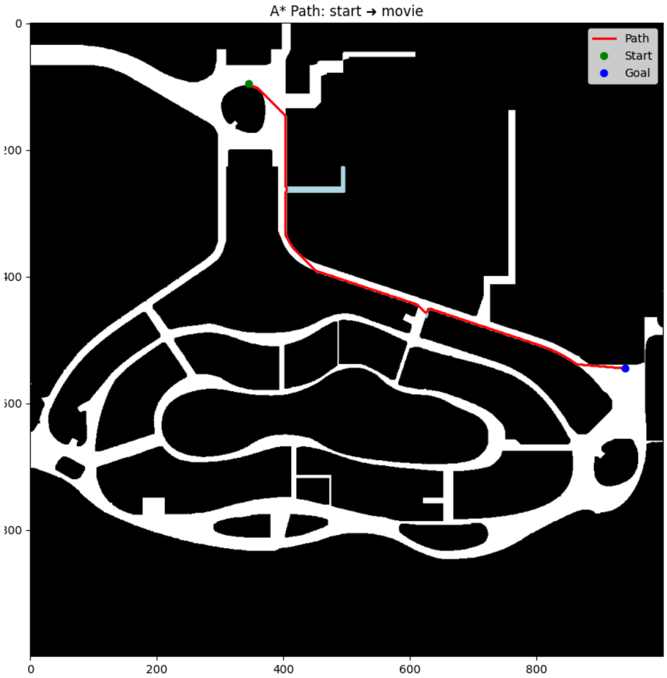
Method	A*
Path	start --- snacks
Total Distance (m)	28.99
Visited Cells	4074
Runtime (s)	0.2631



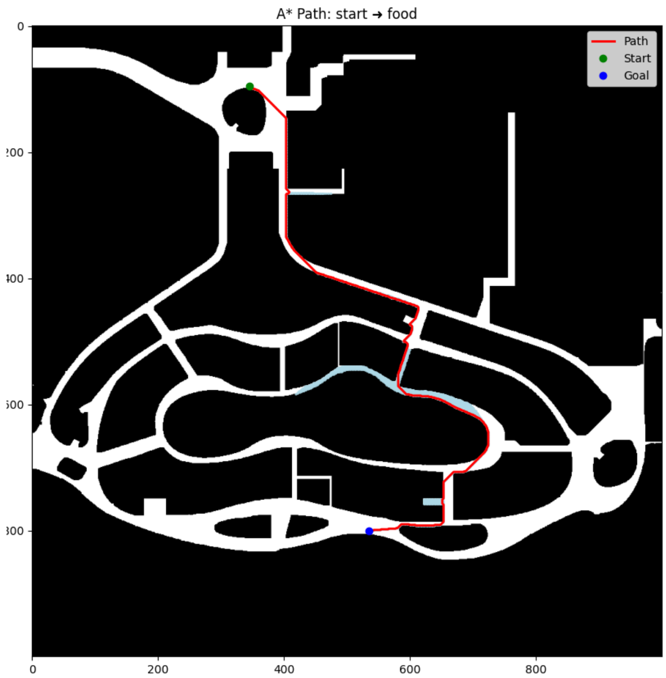
Method	A*
Path	start --- store
Total Distance (m)	32.04
Visited Cells	695
Runtime (s)	0.0288



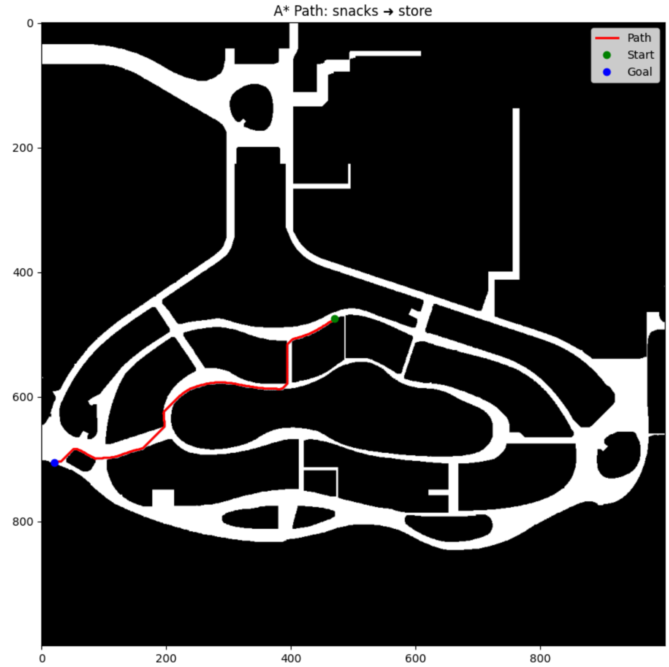
Method	A*
Path	start --- movie
Total Distance (m)	36.14
Visited Cells	1659
Runtime (s)	0.1535



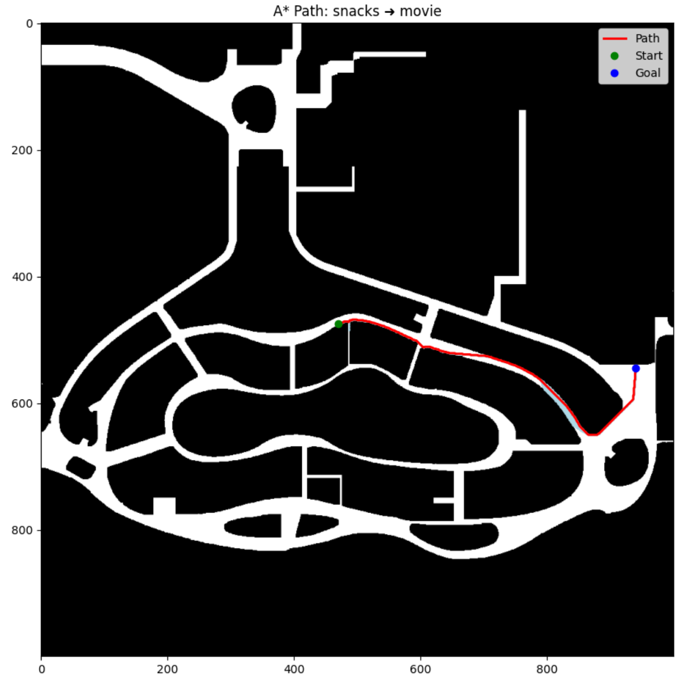
Method	A*
Path	start --- food
Total Distance (m)	46.72
Visited Cells	4302
Runtime (s)	0.4371



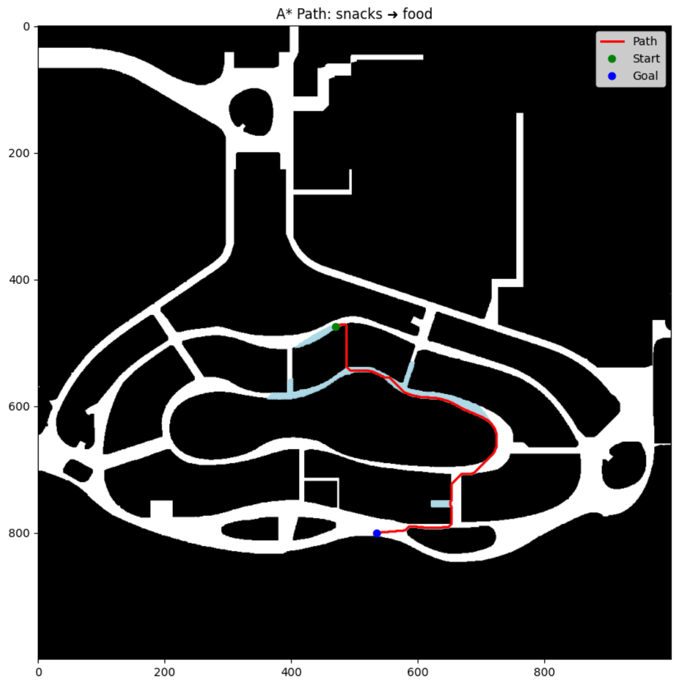
Method	A*
Path	snacks --- store
Total Distance (m)	24.80
Visited Cells	539
Runtime (s)	0.0225



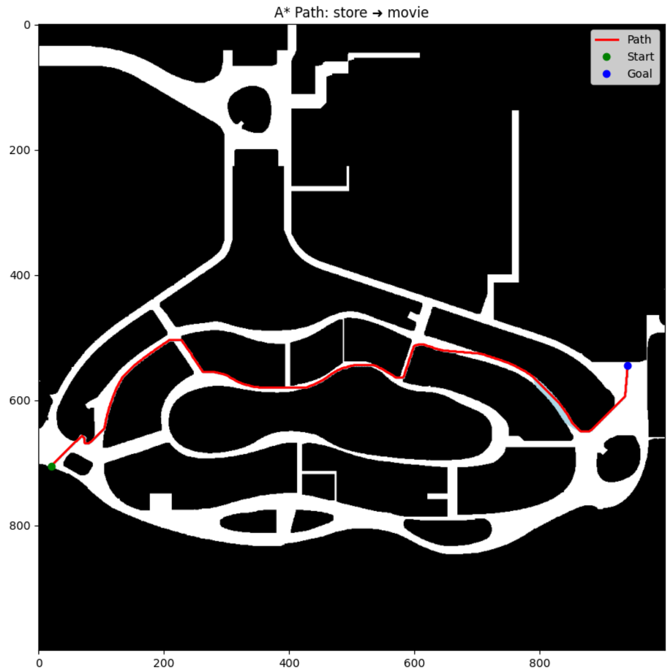
Method	A*
Path	snacks --- movie
Total Distance (m)	25.70
Visited Cells	1335
Runtime (s)	0.1046



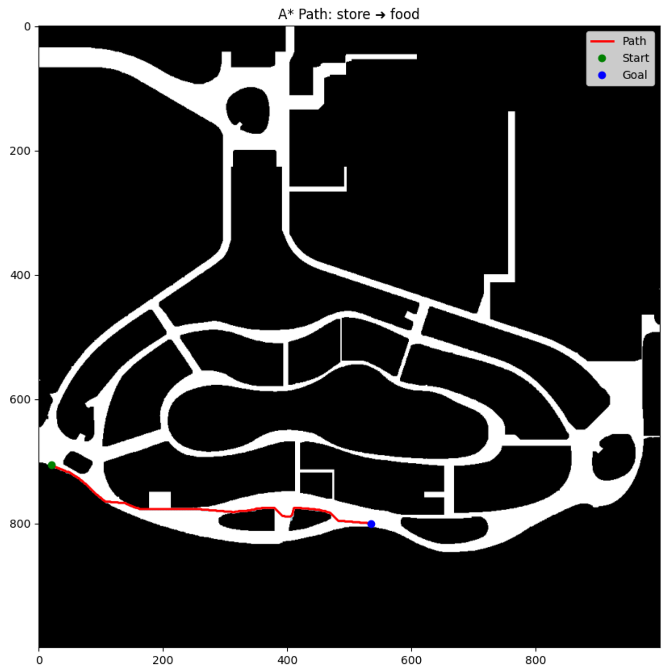
Method	A*
Path	snacks --- food
Total Distance (m)	27.14
Visited Cells	478
Runtime (s)	0.4543



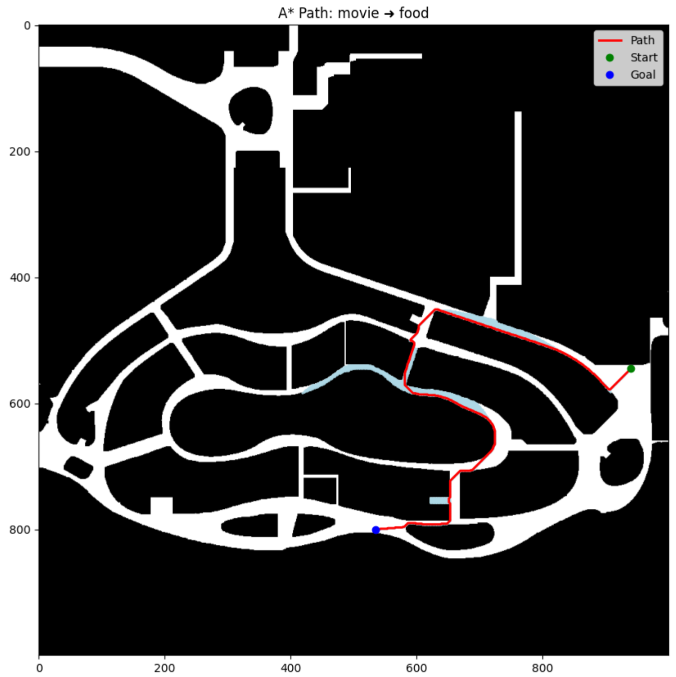
Method	A*
Path	store --- movie
Total Distance (m)	51.42
Visited Cells	1903
Runtime (s)	0.09



Method	A*
Path	store --- food
Total Distance (m)	23.04
Visited Cells	541
Runtime (s)	0.0189



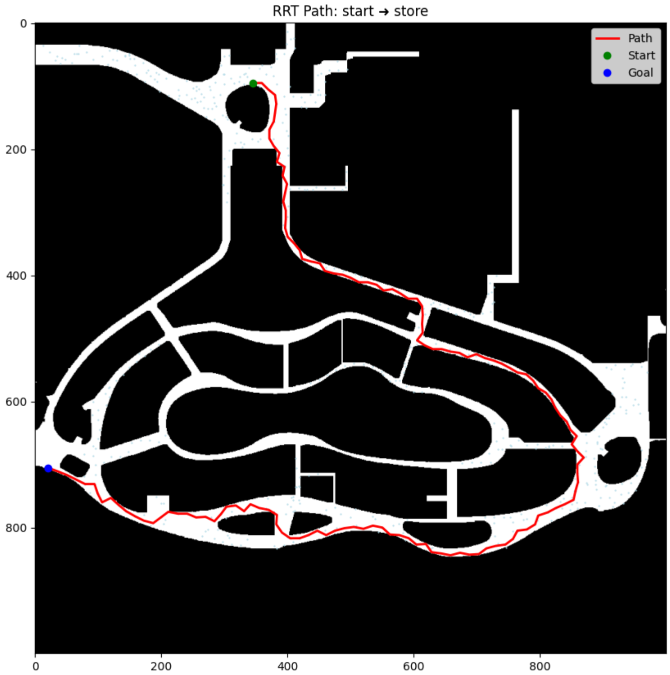
Method	A*
Path	movie --- food
Total Distance (m)	40.37
Visited Cells	5509
Runtime (s)	0.4775



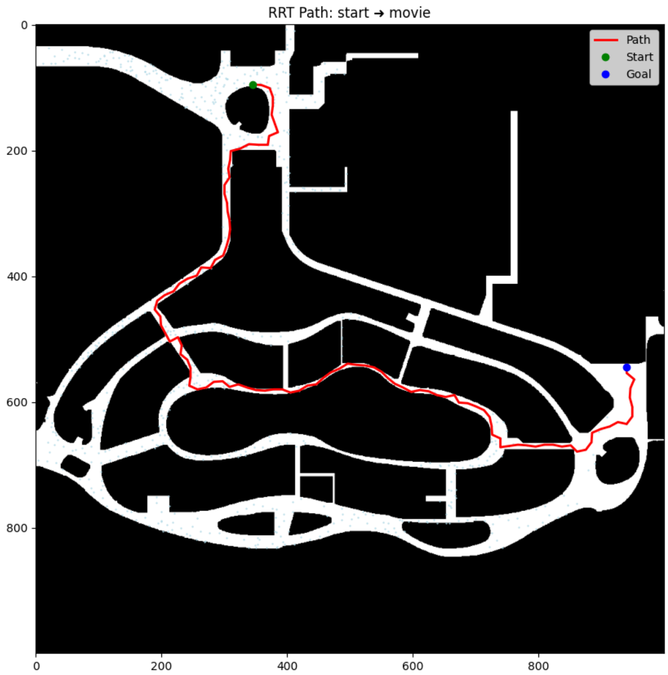
Method	RRT
Path	start --- snacks
Total Distance (m)	155.24
Visited Cells	1191
Runtime (s)	55.6964

Method	RRT
Path	start ---store
Total Distance (m)	403.57
Visited Cells	590
Runtime (s)	13.8701

Method	RRT
Path	start ---store
Total Distance (m)	403.57
Visited Cells	590
Runtime (s)	13.8701



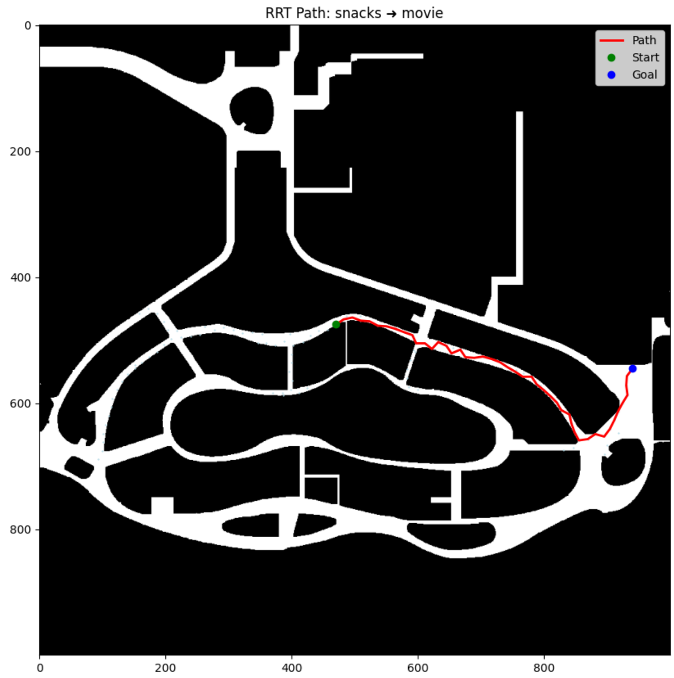
Method	RRT
Path	start --- movie
Total Distance (m)	317.92
Visited Cells	728
Runtime (s)	19.078



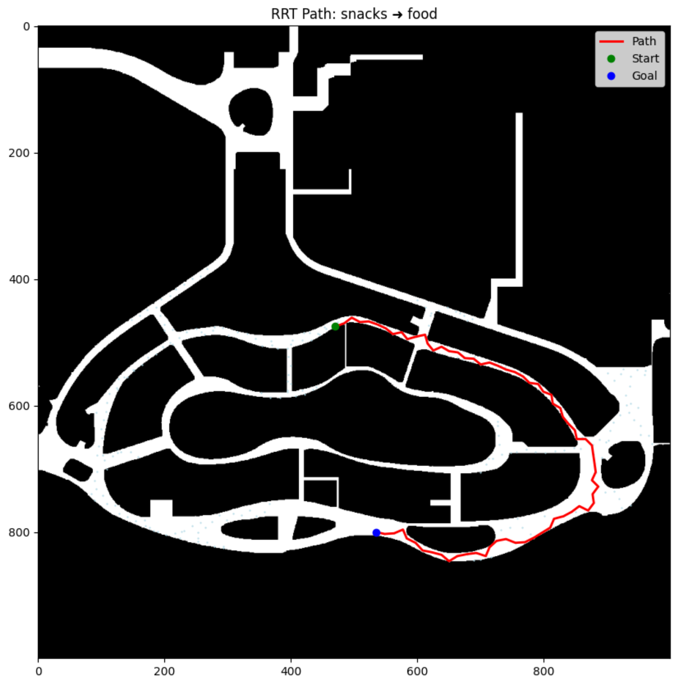
Method	RRT
Path	start --- food
Total Distance (m)	272.20
Visited Cells	425
Runtime (s)	7.8283

Method	RRT
Path	snacks --- store
Total Distance (m)	139.01
Visited Cells	419
Runtime (s)	6.9188

Method	RRT
Path	snacks --- movie
Total Distance (m)	130.5
Visited Cells	118
Runtime (s)	0.611



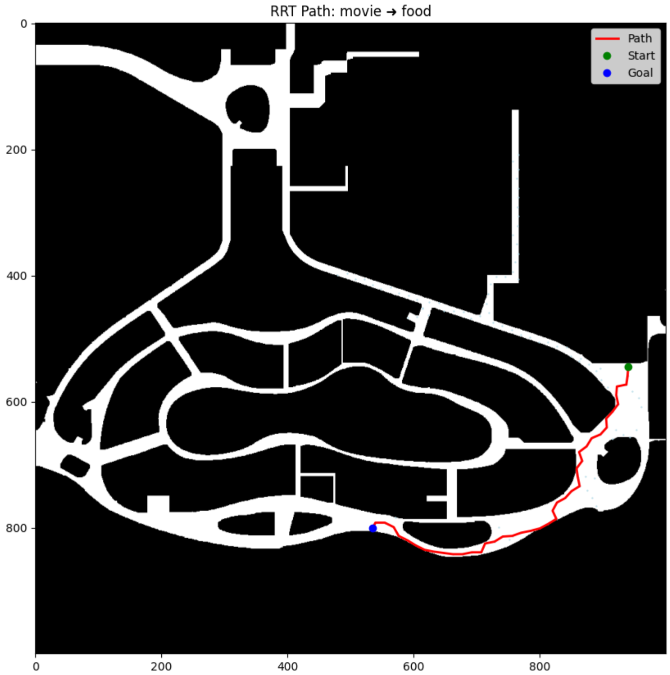
Method	RRT
Path	snacks --- food
Total Distance (m)	205.53
Visited Cells	282
Runtime (s)	2.4363



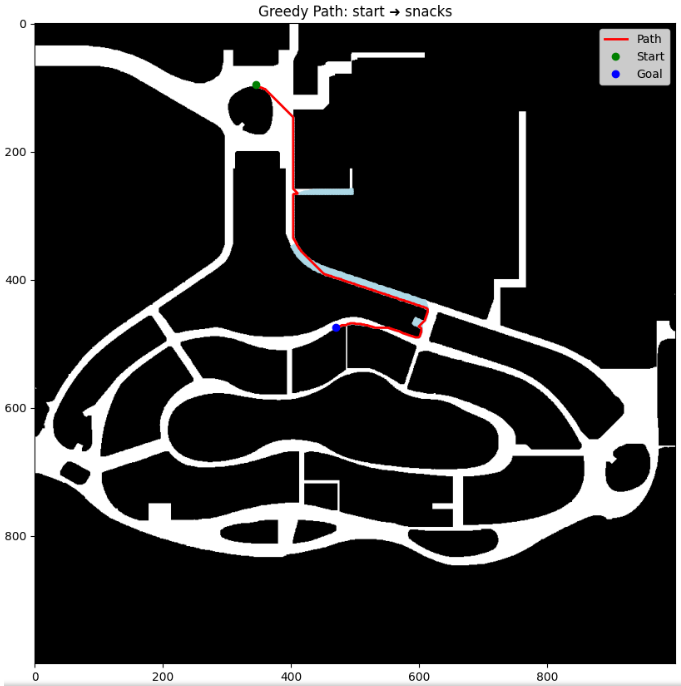
Method	RRT
Path	store --- movie
Total Distance (m)	229.35
Visited Cells	286
Runtime (s)	1.8919

Method	RRT
Path	store --- food
Total Distance (m)	121.97
Visited Cells	580
Runtime (s)	9.2592

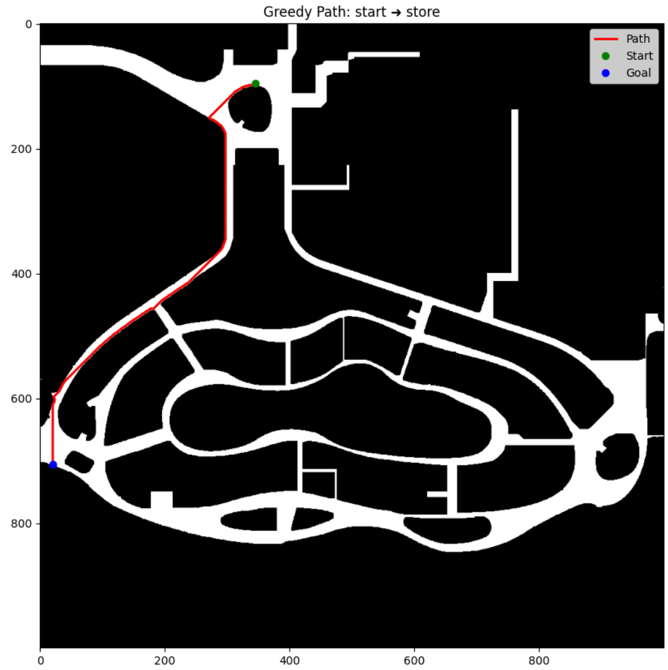
Method	RRT
Path	movie --- food
Total Distance (m)	127.62
Visited Cells	118
Runtime (s)	0.3515



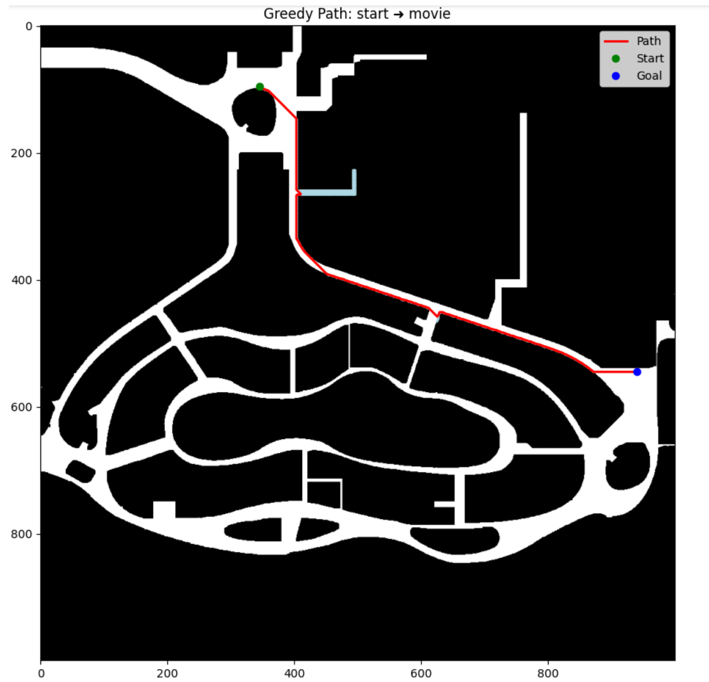
Method	Greedy Path
Path	start -- snacks
Total Distance (m)	29.25
Visited Cells	3815
Runtime (s)	0.0848



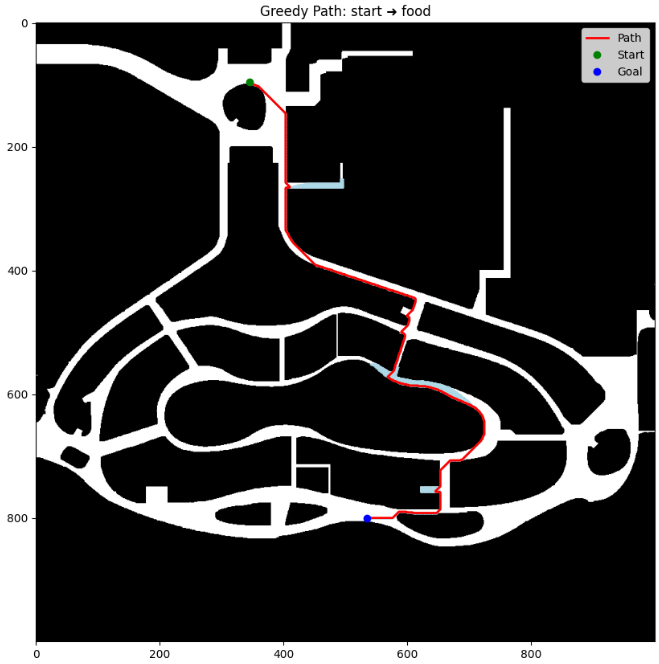
Method	Greedy Path
Path	start -- store
Total Distance (m)	32.15
Visited Cells	677
Runtime (s)	0.0264



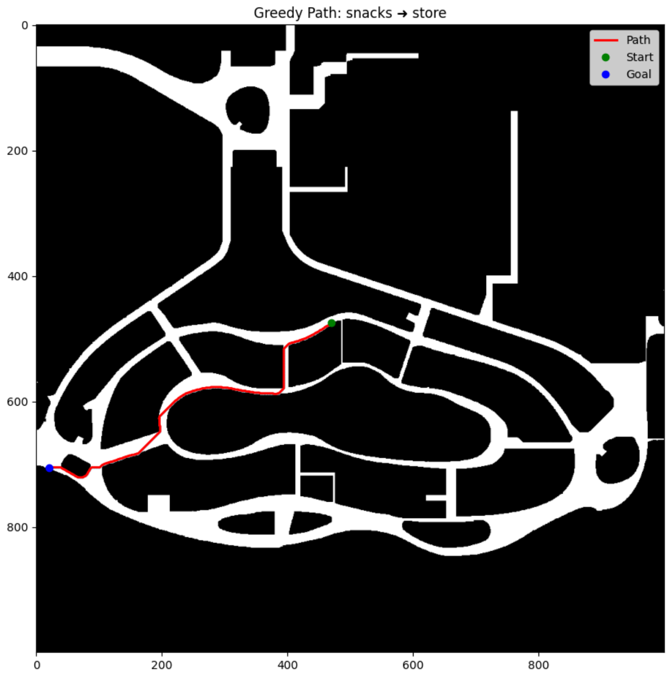
Method	Greedy Path
Path	start -- movie
Total Distance (m)	36.48
Visited Cells	1651
Runtime (s)	0.0391



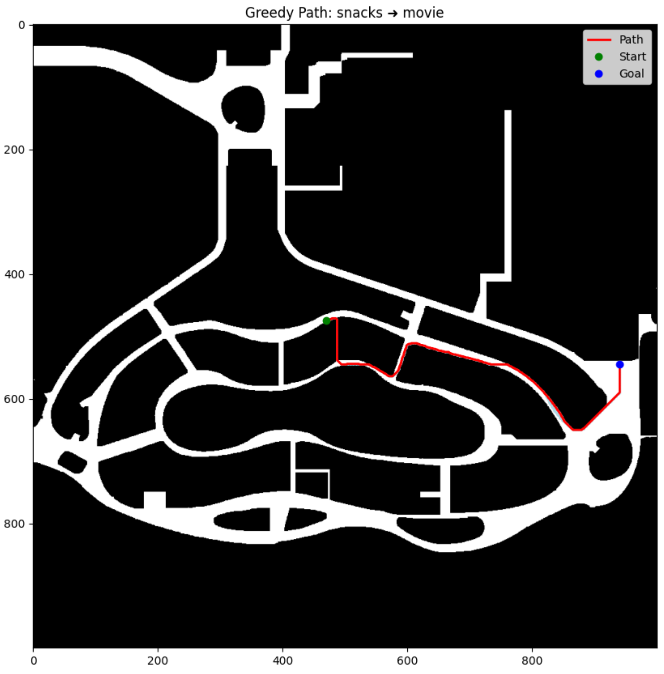
Method	Greedy Path
Path	start -- food
Total Distance (m)	47.97
Visited Cells	3460
Runtime (s)	0.0739



Method	Greedy Path
Path	snacks -- store
Total Distance (m)	24.89
Visited Cells	545
Runtime (s)	0.0199

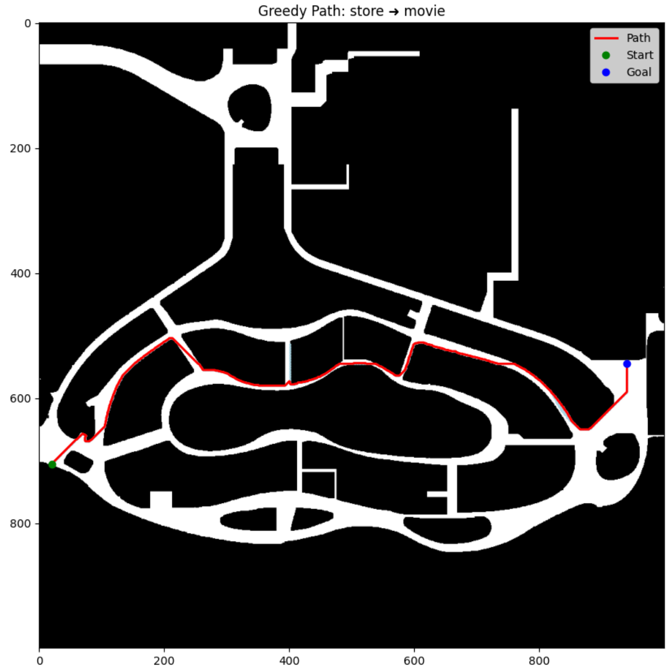


Method	Greedy Path
Path	snacks -- movie
Total Distance (m)	28.98
Visited Cells	843
Runtime (s)	0.0178



Method	Greedy Path
Path	snacks -- food
Total Distance (m)	34.53
Visited Cells	3096
Runtime (s)	0.0794

Method	Greedy Path
Path	store -- movie
Total Distance (m)	51.41
Visited Cells	1364
Runtime (s)	0.064



Method	Greedy Path
Path	store -- food
Total Distance (m)	23.81
Visited Cells	536
Runtime (s)	0.0218

Method	Greedy Path
Path	movie -- food
Total Distance (m)	41.61
Visited Cells	2686
Runtime (s)	0.0872