Algorithms:

1. BFS (Breadth-First Search)

Description:

- BFS is an uninformed search algorithm that explores all nodes at the current depth before moving to the next level.
- It is optimal for unweighted graphs (shortest path in terms of steps, not distance).

Data Structures Used:

- Queue for frontier
- Visited matrix (2D array)
- Parent dictionary or matrix for path reconstruction

2. Dijkstra's Algorithm

Description:

- Dijkstra is a single-source shortest path algorithm for graphs with non-negative weights.
- It guarantees the optimal shortest path based on actual distances.

Data Structures Used:

- Priority Queue (usually a Min-Heap)
- Distance map (dictionary or 2D array)
- Visited set
- Parent dictionary or map for backtracking

3. RRT (Rapidly-Exploring Random Tree)

Description:

- RRT is a **sampling-based** path planning algorithm designed for **continuous spaces**.
- It grows a tree from the start by randomly sampling points and connecting them to the nearest node in the tree, avoiding obstacles.

Data Structures Used:

- **Tree structure** (often represented via a list of nodes with parent pointers)
- **KD-tree** (optional, for nearest neighbor search)
- **List or set** of sampled points

Analysis:

Aspect	BFS	Dijkstra	RRT
Search Type	Uninformed	Informed (uses cost)	Sampling-based
Path Quality	Shortest in steps (grid)	Optimal shortest path (cost-wise)	Sub-optimal, depends on randomness
Time Consumption	Moderate (wide expansion)	High (more computation per node)	Low to moderate (depends on samples)
Memory Usage	High (stores all visited)	Higher (priority queue + distances)	Lower (stores tree only)
Data Structure Complexity	Simple (queue + visited map)	Moderate (heap + map + visited set)	Moderate to complex (tree + NN search)
Best For	Small grids, uniform-cost maps	Weighted maps, deterministic path	Complex, high-dimensional spaces

Experiment Results:

Memory Usage (peak MB)

Start Point : End Point -

Map 1

Algorithm	(0, 0) : (9, 9)	(1, 4) : (9, 0)	(7, 8) : (0, 1)
BFS	0.0060	0.0022	0.0022
Dijkstra	0.0056	0.0032	0.0032
RRT	0.0025	0.0025	0.0025

Map 2

Algorithm	(0, 0) : (14, 14)	(10, 3) : (3, 12)	(7, 13) : (13, 0)
BFS	0.0125	0.0000	0.0070
Dijkstra	0.0125	0.0000	0.0082
RRT	0.0082	0.0001	0.0082

Map 3

Algorithm	(0, 0) : (19, 49)	(5, 35) : (16, 15)	(2, 3) : (12, 14)
BFS	0.0498	0.0291	0.0279
Dijkstra	0.0535	0.0425	0.0401
RRT	0.0506	0.0917	0.0911

Computation Time (seconds)

Start Point : End Point -

Map 1

Algorithm	(0, 0) : (9, 9)	(1, 4) : (9, 0)	(7, 8) : (0, 1)
BFS	0.0000	0.0000	0.0000
Dijkstra	0.0000	0.0000	0.0000
RRT	0.0156	0.0045	0.0000

Map 2

Algorithm	(0, 0) : (14, 14)	(10, 3) : (3, 12)	(7, 13) : (13, 0)
BFS	0.0022	0.0000	0.0010
Dijkstra	0.0023	0.0000	0.0000
RRT	0.0120	0.0000	0.0437

Map 3

Algorithm	(0, 0) : (19, 49)	(5, 35) : (16, 15)	(2, 3) : (12, 14)
BFS	0.0161	0.0030	0.0050
Dijkstra	0.0000	0.0053	0.0025
RRT	1.0191	1.9015	2.1139

Path Length (meters)

Start Point : End Point -

Map 1

Algorithm	(0, 0) : (9, 9)	(1, 4) : (9, 0)	(7, 8) : (0, 1)
BFS	15.6568	14.0000	13.6568
Dijkstra	15.6568	14.0000	13.6568
RRT	25.6568	17.4142	15.4142

Map 2

Algorithm	(0, 0) : (14, 14)	(10, 3) : (3, 12)	(7, 13) : (13, 0)
BFS	26.8284	NP	19.8284
Dijkstra	28.4852	NP	19.8284
RRT	30.7279	NP	38.0710

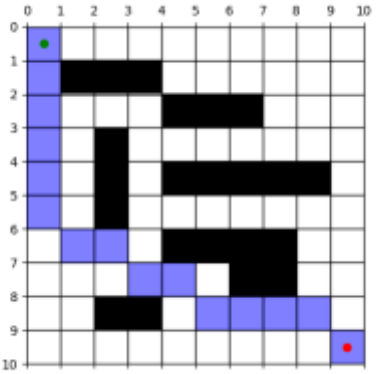
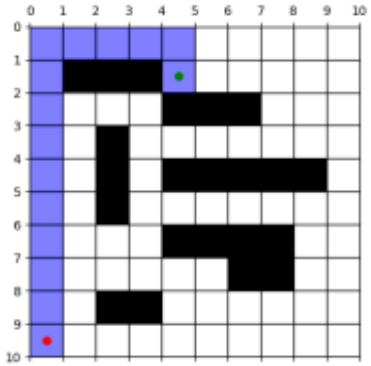
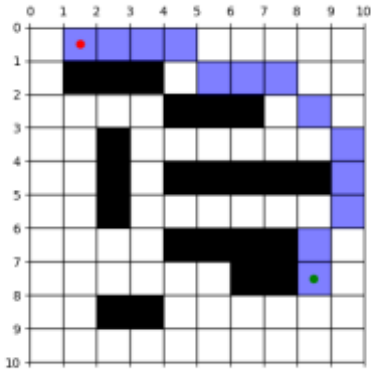
Map 3

Algorithm	(0, 0) : (19, 49)	(5, 35) : (16, 15)	(2, 3) : (12, 14)
BFS	61.2132	35.4852	40.6568
Dijkstra	62.8700	39.6274	41.4852
RRT	78.0416	63.6274	51.4852

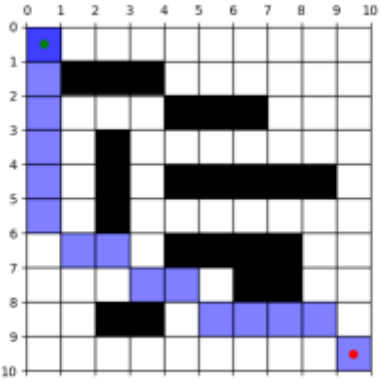
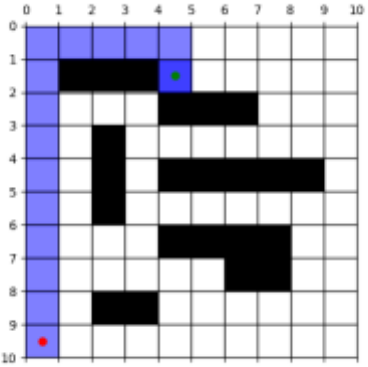
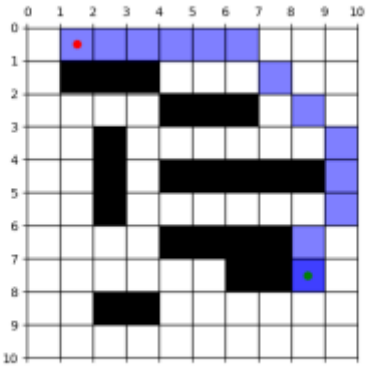
Output Plots:

Map1

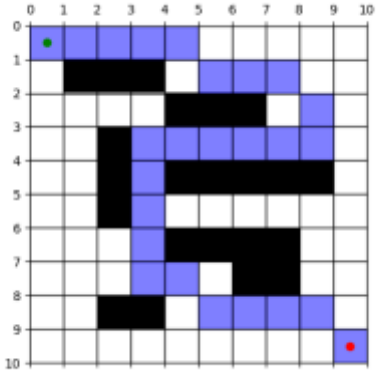
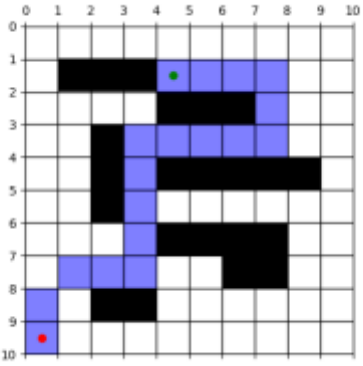
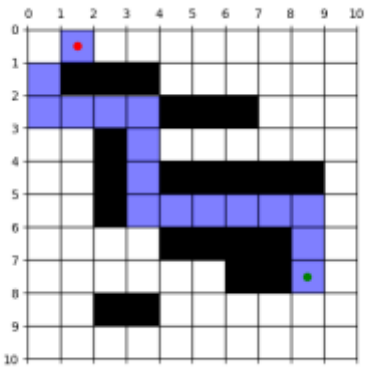
BFS

Dijkstra

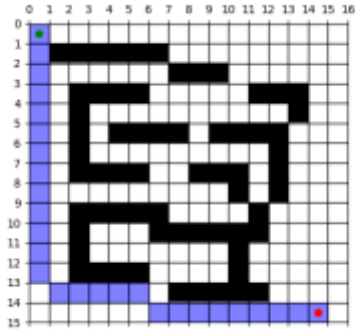
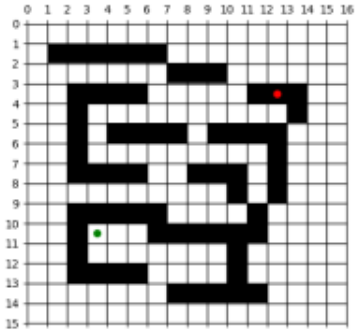
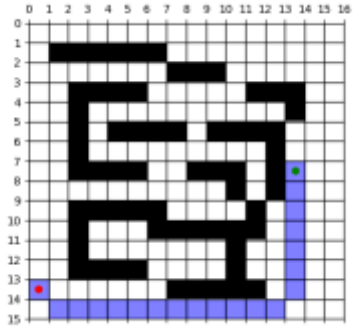




RRT

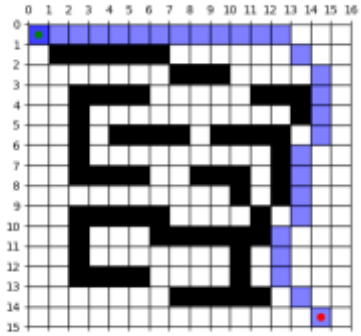
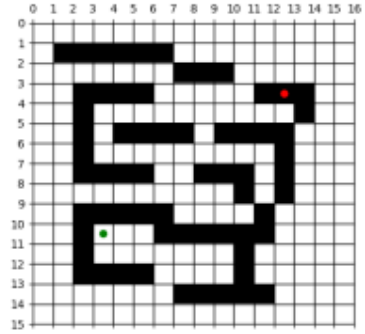
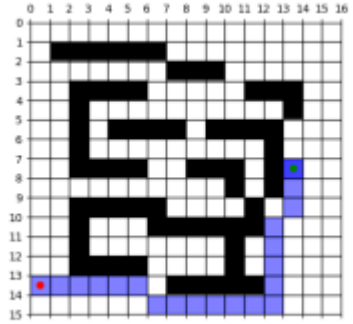




Map2

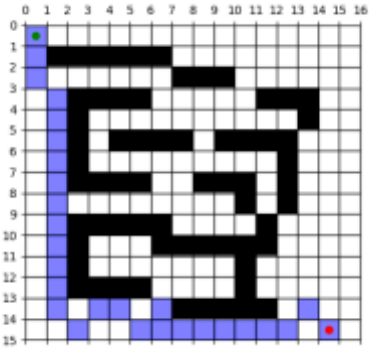
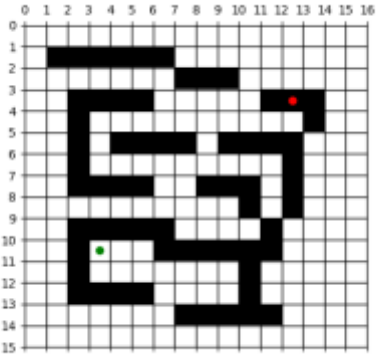
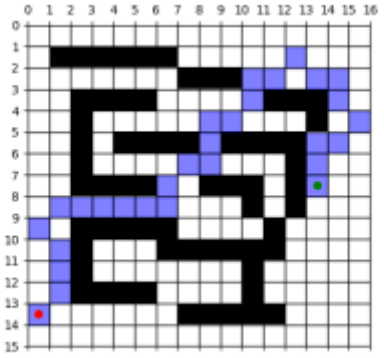
BFS

Dijkstra

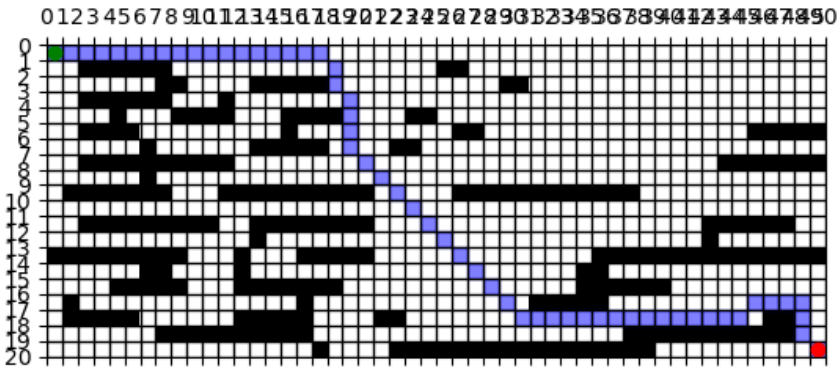




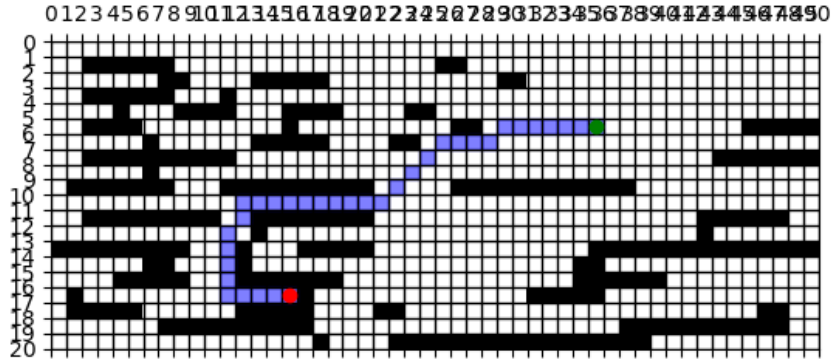
RRT

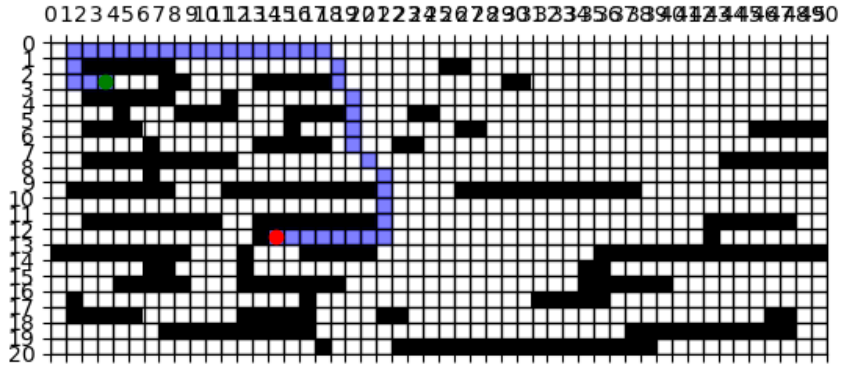




Map3

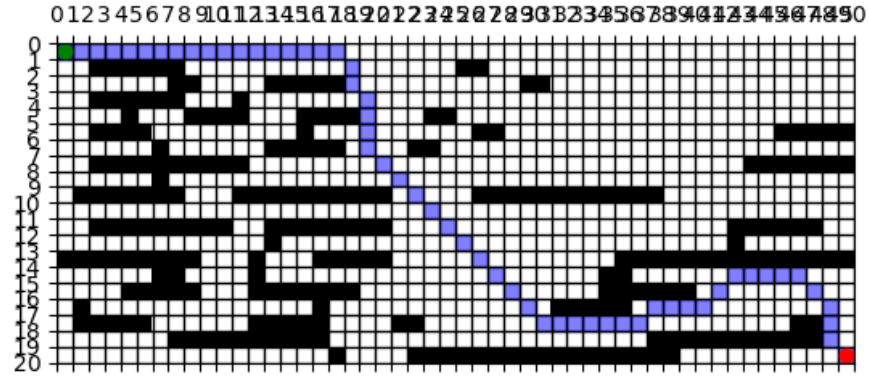
BFS

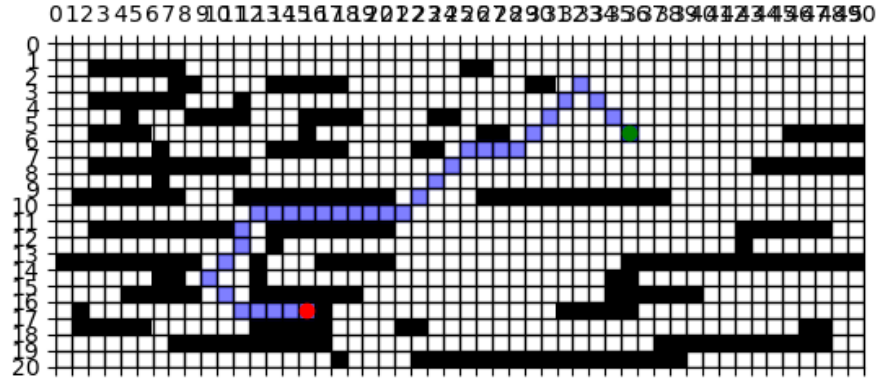


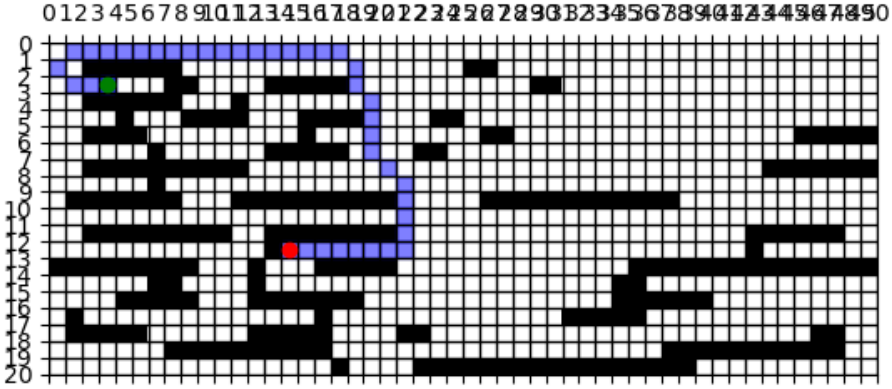




Dijkstra







RRT