Phan Ngọc Hạnh Nhi

2131209002

LAB 2

1. Experimental Results

1.1. Uninformed Search

Maze	Depth-First Search			Breadth-First Search		
	#Nodes Explored	Solution Length	Optimal?	#Nodes Explored	Solution Length	Optimal?
tiny	14	10	No	15	8	Yes
medium	144	130	No	266	68	Yes
big	390	210	Yes	617	210	Yes

1.2. Informed Search

Maze	Best-First Search			A*		
	#Nodes Explored	Solution Length	Optimal?	#Nodes Explored	Solution Length	Optimal?
tiny	15	8	Yes	15	8	Yes
medium	269	68	Yes	269	68	Yes
big	620	210	Yes	620	210	Yes

2. Short Discussion

2.1. Optimality

- Breadth-First Search (BFS) and A* always produce the shortest path because they expand nodes based on path cost (or cost plus heuristic).
- Greedy Best-First Search can quickly find good paths but is not guaranteed to be optimal if the heuristic is inaccurate.
- Depth-First Search (DFS) may sometimes find the optimal solution, but this is coincidental—it does not systematically search for minimal cost.

2.2. Efficiency (Nodes Explored)

- BFS explores the most nodes, as it expands equally in all directions until the goal is found.

- A* explores fewer nodes because the heuristic helps focus the search toward the goal.
- Best-First Search usually expands the fewest nodes when the heuristic is effective, but can perform poorly if the heuristic misleads the search.
- DFS explores fewer nodes in small mazes but may waste effort on deep, unproductive paths.

2.3. Trade-offs

Algorithm	**Advantages**	**Limitations**
DFS	Simple implementation, low memory usage	Not optimal, may explore irrelevant paths
BFS	Always finds the shortest path	Slower and memory-intensive
Best-First Search	Fast, efficient with good heuristics	May fail to find optimal paths
A*	Balances accuracy and efficiency	Dependent on heuristic quality