CSE537 – Artificial Intelligence
Project 1 – Report

Rajesh Prabhakar **SBU ID: 112872762**
Udit Gupta **SBU ID: 112715403**

Question 1 – Stats

Python command	python pacman.py -l tinyMaze -p SearchAgent
Search nodes expanded	15
Running time	0.2 seconds
Memory usage	42.730 MB
Total cost of path	10
Score	500.0

Python command	python pacman.py -l mediumMaze -p SearchAgent
Search nodes expanded	146
Running time	2.4 seconds
Memory usage	65.945 MB
Total cost of path	130
Score	380

Python command	python pacman.py -l bigMaze -z .5 -p SearchAgent
Search nodes expanded	390
Running time	5.9 seconds
Memory usage	55.938 MB
Total cost of path	210
Score	300

Question 2 – Stats

Python command	python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
Search nodes expanded	269
Running time	0.0 seconds
Memory usage	65.801 MB
Total cost of path	68
Score	442

Python command	python pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5
Search nodes expanded	620
Running time	0.0 seconds
Memory usage	55.891 MB
Total cost of path	210
Score	300.0

Question 3 – Stats

Python command	python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
Search nodes expanded	269
Running time	0.0 seconds
Memory usage	65.734 MB
Total cost of path	68
Score	442.0

Python command	python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
Search nodes expanded	186
Running time	0.0 seconds
Memory usage	65.605 MB
Total cost of path	1
Score	646.0

Python command	python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
Search nodes expanded	108
Running time	0.0 seconds
Memory usage	66.965 MB
Total cost of path	68719479864
Score	418.0

Question 4 – Stats

Python command	python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
Search nodes expanded	549
Running time	0.0 seconds
Memory usage	55.789 MB
Total cost of path	210
Score	300.0

Question 5 – Stats

Python command	python pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
Search nodes expanded	252
Running time	0.0 seconds
Memory usage	43.211 MB
Total cost of path	28
Score	512.0

Python command	python pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
Search nodes expanded	1966
Running time	0.2 seconds
Memory usage	57.105 MB
Total cost of path	106
Score	434.0

Question 6 – Stats

Python command	python pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5
Search nodes expanded	1136
Running time	0.1 seconds
Memory usage	45.062 MB
Total cost of path	106
Score	434.0

Question 7 – Stats

Python command	python pacman.py -l trickySearch -p AStarFoodSearchAgent
Search nodes expanded	11632
Running time	13.5 seconds
Memory usage	89.539 MB
Total cost of path	60
Score	570

Analysis:

As seen from the stats above the DFS algorithm takes 2.4 and 5.9 seconds for medium and big maze respectively whereas the BFS algorithm takes 0.0 seconds so for time constraint applications it is better to go for BFS.

The BFS expands 269 and 620 nodes for medium and big maze respectively whereas the DFS expands 146 and 390 nodes, so for applications where time is not a constraint DFS can be implemented given that there is a definite solution at finite depth d.

DFS takes the cost of 130 for medium maze while BFS takes just a cost of 68 which means the BFS takes the optimal path.

UCS expands 269 nodes and has a cost of 68 for medium maze which same as BFS (taking each edge cost as 1).

A* expands 549 nodes and has a cost of 210, so A* performs better than BFS as it maintains the same cost as BFS but with lesser number of nodes expanded. A* expands lesser nodes because it has heuristic function which determines which nodes to expand next.