Question 1: depthFirstSearch

For this problem, we used the stack data structure for providing the needed LIFO property. Also, there is a dictionary which helps to check if the node is visited or not. If the node is not visited, then the node gets added to the stack. In the end a list of actions is returned which pacman uses to traverse.

Type	Number of nodes expanded	Memory usage	Running Time (Second)	Critical Analysis (Cost)
TinyMaze	146	$O(\log(n))$	0	130
MediumMaze	146	$O(\log(n))$	0	130
LargeMaze	390	$O(\log(n))$	0	210

Question 2: breathFirstSearch

For this problem, we used the queue data structure for providing the needed FIFO property. Also, there is a dictionary which helps to check if the node is visited or not. If the node is not visited, then the node gets added to the stack. In the end a list of actions is returned which pacman uses to traverse.

Type	Number of nodes expanded	Memory usage	Running Time (Second)	Critical Analysis (Cost)
MediumMaze	269	O(n)	0	68
LargeMaze	620	O(n)	0	210

Question 3: uniformCostSearch

In this, 'PriorityQueue' data structure is used. As Priority Queue helps in assigning the priority values. And based on decreasing priority values, it will complete the path. Here, Expanded list will store the states which are visited. The movement of the pacman takes place by checking the state of the node using 'getSuccessor' method. Finally, when the goal is reached, 'actions' is returned.

Type	Number of nodes expanded	Memory usage	Running Time (Second)	Critical Analysis (Cost)
MediumMaze	269	$O(b^{(1+\text{floor}(C/\varepsilon))})$	0.1	68
MediumDottedMaze	186	$O(b^{(1+\text{floor}(C/\varepsilon))})$	0	1
MediumScaryMaze	108	$O(b^{(1+\text{floor}(C/\varepsilon))})$	0	68719479864

Question 4: aStar

In this, 'PriorityQueue' data structure is used. Along with 'heuristic' function and 'Expanded' list. Heuristic function, gets the minimum path. Here, Expanded list will store the states which are visited.

Type	Number of nodes expanded	Memory usage	Running Time (Second)	Critical Analysis (Cost)
BigMaze	549		0.3	210

Question 5: Corners problem

Initially, the state contains two parts. State[0] = starting position and State[1]=List of corners. For encoding, the state is been checked. If all the positions are reached; then, the length of the list will be zero and the goal is achieved.

Type	Number of nodes expanded	Memory usage	Running Time (Second)	Critical Analysis (Cost)
TinyCorners	252	$O(b^d)$	0	28
MediumCorners	1966	$O(b^d)$	0	103

Question 6: Corners problem in CornersHeuristic

The heuristic function which is used is the minimum of the sum of Manhattan distances from source to all possible combinations of the corners. We have used this heuristic function, because it calculates the direct shortest distance for all possible paths.

The heuristic function used, is admissible because, f(n) from the starting state to the goal state is shortest when we consider the direct distance between the two points.

And if all the possible paths are considered from starting state to the goal state, we get the f(n) <= other f(n)'s. Hence, the function is consistent.

Note: The formula used is f(n) = g(n) + h(n);

Type	Number of nodes expanded	Memory usage	Running Time (Second)	Critical Analysis (Cost)
testSearch	741	O(l+b)	0.2	106

Question 7: Food heuristic

The heuristic function which is used is Manhattan distance. We have used this heuristic function, because it calculates the direct shortest distance between the pacman and the food.

We have considered the shortest distance of the pacman to the food. Then, the pacman traverse to the nearest food and so on; till it finally eats all the food.

The heuristic is admissible, because at ever state of time, it calculates the minimum Manhattan distance between the pacman and the food. Also, the algorithm is consistent, because in most of the situations, f(n) is less than other f(n)'s calculated from the same state.

Note: The formula used is f(n) = g(n) + h(n);

Type	Number of nodes expanded	Memory usage	Running Time (Second)	Critical Analysis (Cost)
trickySearch	11975	O(l+b)	26.9	60