REPORT
PROJECT 1 : The Searchin' Pac-Man

Aashray Arora (aashray.arora@stonybrook.edu)
Eshita Bheda (eshita.bheda@stonybrook.edu)
Varun Shastry (varun.shastry@stonybrook.edu)

INTRODUCTION

This report gives a description and summary of the implementation and characteristics of each of the search techniques and heuristics used as a part of this project.

DESCRIPTION

Question 1 : Depth First Search (DFS)

- Implemented Depth First Graph Search using stack as a fringe.

Question 2 : Breadth First Search

- Implemented Breadth First Search using Queue as a fringe.

Question 3 : Uniform Cost Search

- Implemented Uniform Cost Search using a priority queue as a fringe. Priority is given by cost.

Question 4 : A search*

- Implemented A* search using a priority queue as fringe (for cost) and allowed for using the value returned by a specified heuristic.

Question 5 : Corners Problem

- Implemented *getStartState, isGoalState and getSuccessors.*
- Added visited corners to the state representation. Kept track of which corners are visited. Mark corner as visited when it is reached in the *getSuccessor* function.

Question 6 : Corners Heuristic

- Implemented a heuristic that returns a value which is the sum of manhattan distance to the closest unvisited corner and manhattan distance from that corner to next unvisited corner and so on, until we have taken into account all unvisited corners.

Question 7 : Food Heuristic for eating all Dots

- Implemented a heuristic that returns a value which is the sum of the manhattan distances to the closest and farthest food.

STATISTICS

Question	No.of nodes expanded	Running Time (seconds)	Cost
1 : Depth First Graph Search	tinyMaze: 14 mediumMaze: 144 bigMaze: 390	tinyMaze: 0.0 mediumMaze: 0.0 bigMaze: 0.0	tinyMaze: 10 mediumMaze: 130 bigMaze: 210
2 : Breadth First Graph Search	mediumMaze: 269 bigMaze: 620	mediumMaze: 0.0 bigMaze: 0.0	mediumMaze: 68 bigMaze: 210
3 : Uniform Cost Search	mediumMaze: 269 mediumDottedMaze, StayEastSearchAgent: 186 mediumScaryMaze, StayWestSearchAgent: 108	mediumMaze: 0.0 mediumDottedMaze, StayEastSearchAgent: 0.0 mediumScaryMaz, StayWestSearchAgent: 0.0	mediumMaze: 68 mediumDottedMaze, StayEastSearchAgent: 1 mediumScaryMaze, StayWestSearchAgent: 68719479864
4 : A* search *(bigMaze, manhattan heuristic)*	549	0.3	210
5: Corners Problem	tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem: 252 mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem: 1966	tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem: 0.0 mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem: 0.2	tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem: 28 mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem: 106

6: Corners Heuristic *(mediumCorners -p AStarCornersAgent -z 0.5)*	692	0.2	106
7: Food Heuristic - Eating all Dots *(trickySearch -p AStarFoodSearchAgent)*	8441	14.1	60

CONCLUSION

Tree search ensures we do not re-visit a state, thus improving running time. DFS graph search is complete but not optimal, we can see it does not return the optimal solution for the medium maze as BFS does. BFS is complete and optimal (only if all costs are same), since it does not take into account the cost, when costs are different it is not optimal to find the least cost solution. BFS finds the shortest path in terms of number of actions. Uniform cost search is complete and optimal, it finds the least cost path, but it expands too many nodes, thus increasing running time. These were uninformed search techniques. A* search is informed. Other than cost, it also prioritizes based on a heuristic (prediction), that tells us if we are getting close to a goal state. A* expands lesser nodes. For A* graph search, the heuristic needs to be consistent (not only admissible) for A* graph search to be optimal. In the corners problem. Maintaining the visited corners helps. The corners heuristic gives a good result. The method used is described in the introduction. The Food heuristic also gives a good result, the method used is described in the introduction.