CSCE4613/5613 – Artificial Intelligence

Fall 2021 Assignment 2

Name: Rashmitha Thoranala

Mail-id: rthorana@uark.edu

Id: 010962830

Implementation:

I implemented the Depth first search, Breadth first search, Uniform cost search and Astar search in the search.py file of Pacman game assignment and enable Pacman agent to find the path on maze to reach his food. The searching function will return a list of actions that leads the Pacman from the start position to his food. The cost of this game is the total cost step of each action performed during the path. We will test all the searching functions implemented on three different kinds of maze namely tiny maze, medium maze and big maze.

- I used Stack data structure that is available in the util.py for the implementation of DFS and able to find the path of Pacman agent from the start position to reach the food position.
- I used Queue data structure that is available in the util.py for the implementation of BFS and able to find the path of Pacman agent from the start position to reach the food position.
- I used priority queue data structure that is available in the util.py for the implementation of UCS and able to find the path of Pacman agent from the start position to reach the food position.
- I used priority queue data structure that is available in the util.py for the implementation of ASTAR and able to find the path of Pacman agent from the start position to reach the food position. Here I also use Manhattan heuristic function to calculate the total cost.

Testing commands:

For Depth First Search we can test on three mazes by using following commands:

- python pacman.py -l tinyMaze -p SearchAgent -a fn=dfs
- python pacman.py -l mediumMaze -p SearchAgent -a fn=dfs
- python pacman.py -l bigMaze -p SearchAgent -a fn=dfs

For Breadth First Search we can test on three mazes by using following commands:

- python pacman.py -l tinyMaze -p SearchAgent -a fn=bfs
- python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
- python pacman.py -l bigMaze -p SearchAgent -a fn=bfs

For Uniform Cost Search we can test on three mazes by using following commands:

- python pacman.py -l tinyMaze -p SearchAgent -a fn=ucs

- python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
- python pacman.py -l bigMaze -p SearchAgent -a fn=ucs

For Astar Search we can test on three mazes by using following commands:

- python pacman.py -l tinyMaze -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
- python pacman.py -l mediumMaze -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
- python pacman.py -l bigMaze -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic