1. The source code contains the implementation of the Bidirectional search algorithm for the Pacman game adapted from the CS 188 course in UC Berkeley.

2. The source code contains two new algorithms MM0 and MM. Both are bidirectional search algorithms guaranteed to meet in the middle implemented in search.py.

3. MM0 algorithm runs BFS search from both start position which is forward search, and from goal node, which is backward search.

4. Similarly MM algorithm runs an A\* search from both start and goal node and meets in the middle.

5. MM algorithm uses a bidirectional heuristic implemented in searchAgents.py, which returns manhattan distance from goal position in forward search and distance from start node in backward search.

6. Prerequisites:- python3

7. There are eight different layouts tiny, small, medium, big, open, contours, and two custom mazes with various starts positions totaling up to 20 mazes.

8. All the layouts can be found in the layouts subdirectory

9. To run the MM algorthim on tiny maze the command is

python pacman.py --layout tinyMaze -p SearchAgent -a fn=mm,heuristic=bidirectionalHeuristic

For small maze run

python pacman.py --layout smallMaze -p SearchAgent -a fn=mm,heuristic=bidirectionalHeuristic

For small maze with start position run

python pacman.py --layout smallMaze\_1 -p SearchAgent -a fn=mm,heuristic=bidirectionalHeuristic

For big maze run

python pacman.py --layout bigMaze -z 0.5 -p SearchAgent -a fn=mm,heuristic=bidirectionalHeuristic

Similarly to run MM0 algorthim on small maze run

python pacman.py --layout smallMaze -p SearchAgent -a fn=mm0

10. The numbers of search nodes expanded for all algorithms BFS, DFS, UCS, Astar, MM0 and MM on all 20 layouts can be found in results.xlsx