AI - 2 - Ali Ul Haq - 500667810

Assignment 1

1.a):Is cycle detection required when using:

- DFS:
    - 8 Puzzle problem
    - Pacman Route planning

For the depth first cycle detection is needed because there is a possibility that DFS can get trapped in cycles of the problem graph and loop forever without finding a solution at all. It will therefore get stuck in one branch of the graph.

- BFS:
    - 8 Puzzle problem
    - Pacman Route planning

For Breadth first there is no need for cycle detection because it will explore each branch before exploring the next frontier for these problems.

1.b): Is cycle detection more important with BFS or DFS, or just as important,

- when solving the 8-puzzle problem?
- when solving the pacman route planning?
- Motivate your answers.

It is more important for Depth First search, as stated in the reasoning in question 1a. Depth first can get into a infinite loop. For the 8 puzzle problem Breadth first would have no need for cycle detection because the branches and structure won't be that large, with the pacman route planning there might be a need for cycle detection because if the goal state is deep in the tree, the amount of memory needed could be very large and its possible that some states may be explored twice

1.c): Explain how the search pattern would look like if you would implement a Breadth First Search to guide Pacman from the start to the goal in the mediumMaze map. Which nodes would be visited when and why?

The search pattern would be that pacman would visit all nodes leading to goal states and the branch or path that reaches the goal state first is chosen as path from start to the goal and all other nodes would be left unexplored.

1.d): Would BFS always find the shortest route to the finish? Motivate your answer.

Yes, because it simply explores each frontier and expands the nodes in frontier until it reaches a goal, it will then find the shortest path. But for this path it is assumed that the cost is equal to 1 for bfs.

1.e): Will BFS's search process will come across every possible position on the board before terminating? If so, why? If not, what nodes will not be part of the BFS' search?

No because for example in the pacman mediummaze problem the whole left side of the board might not be searched because the goal state (solution) is explored/reached first before the BFS's search process reaches that far in the tree.

1.f) Do an estimation about the maximum number of open nodes at the frontier (open list) during the route. How did you come up with this number?

For mediummaze the number would be about

1.h) Did your search algorithm find the route you were expecting? Motivate your Answer. Yes, it found the shortest path to the goal state, for mediummaze it did not take the right spiraling path, which was nearest to the initial state.

Assignment 2

2.a) Explain how the search pattern would look like if you would implement a Depth First Search to guide Pacman from the start to the goal. Which nodes would be visited when and why?

Depth first would explore the most left branch and node to the goal state, which in this case would be the path that goes to the west and continues to the spiral that exist on the left side for mediummaze.

2.b) Program the depthFirstSearch function in the search.py file. I did not expect the depthfirstsearch to go the right side of the maze first and then continue all the way to the left.

2.c) Does the order in which the successor nodes (children) are provided by the function getSuccessorNodes make a large difference?

- Can you speed up the search process by resorting them?
- What does this do to the route?
- Is this fair? Why (not)?

Yes because if you would expand nodes and sort them to the nodes which are nearest to the goal state you would find the goal state earlier when using depthfirst. This is unfair because depth first would find the shortest path without resorting.