Week 4

Progress:

We’ve been successful in solving the first four questions of the problem statement. We’re supposed to find the path to a certain position using different algorithms namely DFS, BFS, Uniform Cost search and A\*.

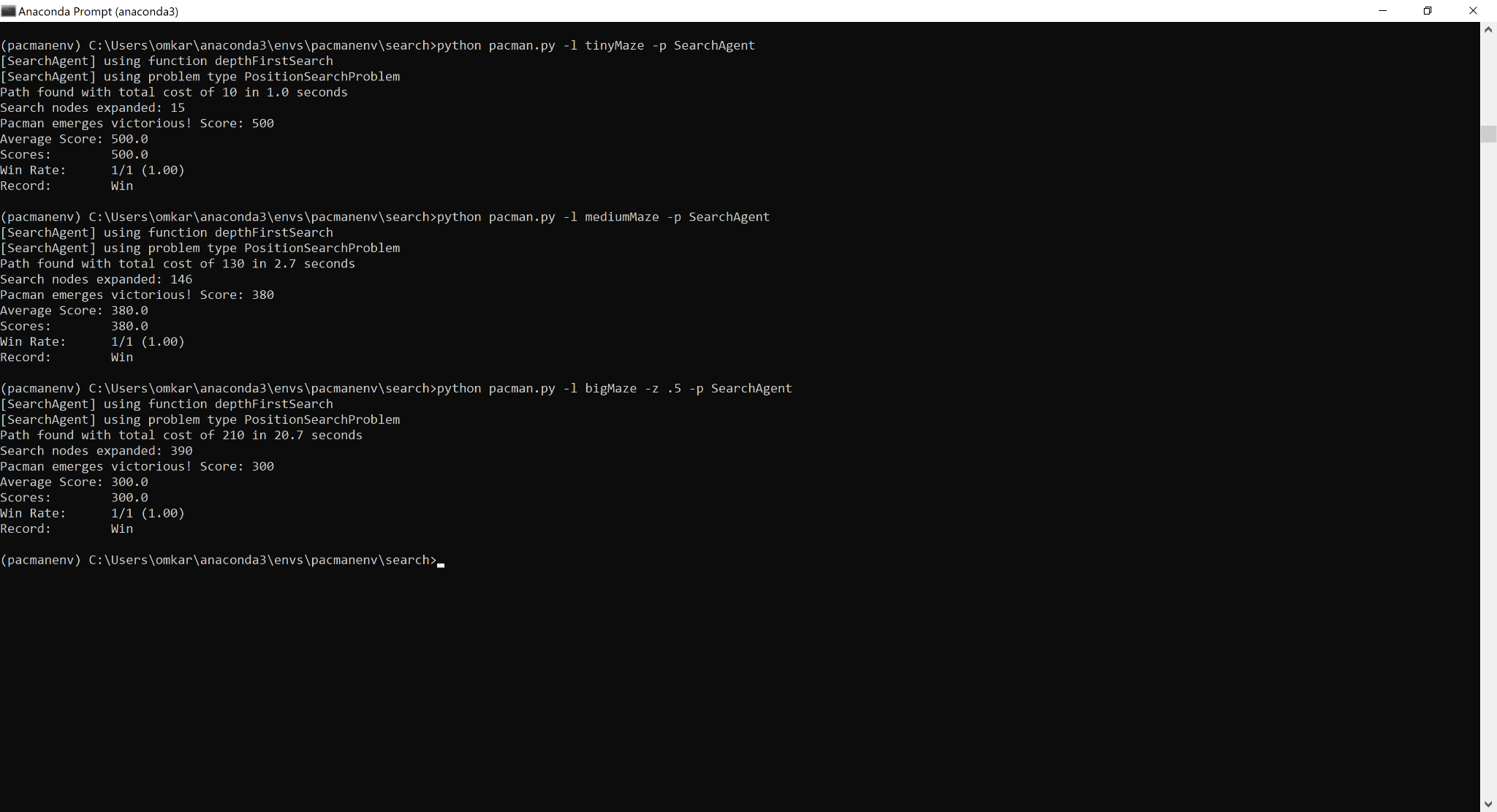
Learnings:

We observed that UCS and A\* gave minimum cost as they consider the cost of every move and then choose which neighbor to explore. Whereas DFS just pops the last pushed node and hence goes on exploring a path until it either reaches the goal or the end of path and ends up with a non-optimal solution.

File Listings and Description:

search.py: It consists of functions that define the algorithm the pacman uses to traverse through the maze to reach a particular position in the maze. We’ve implemented the predefined 4 functions for each algorithm.

**Question 1:**



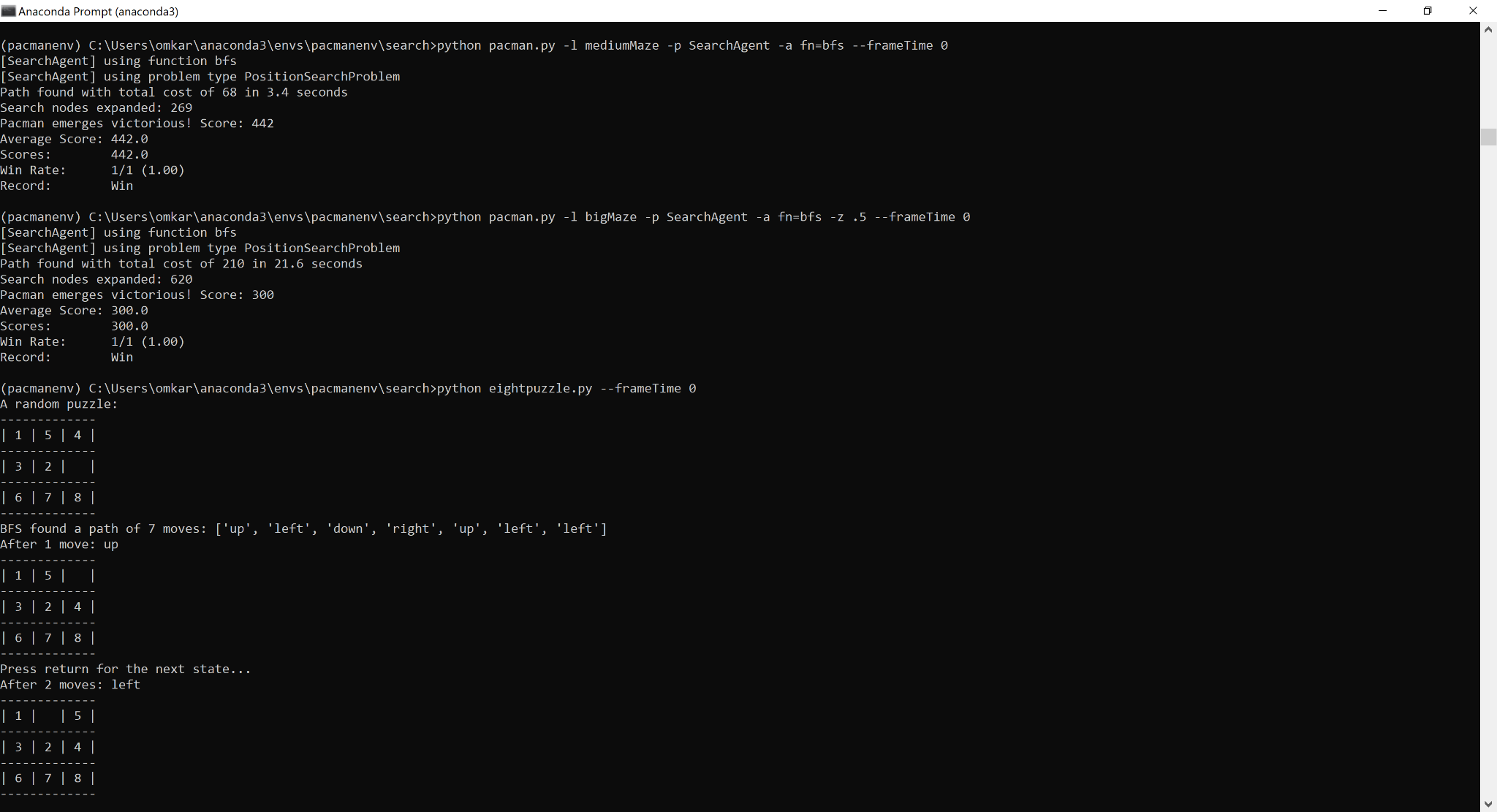
Q: Is the exploration order what you would have expected? Does Pacman actually go to all the explored squares on his way to the goal?

A: Yes the exploration order matches with what we expected for Depth First Search. The pacman doesn’t actually go to all the explored squares in cases it finds a solution before the stack is emptied.

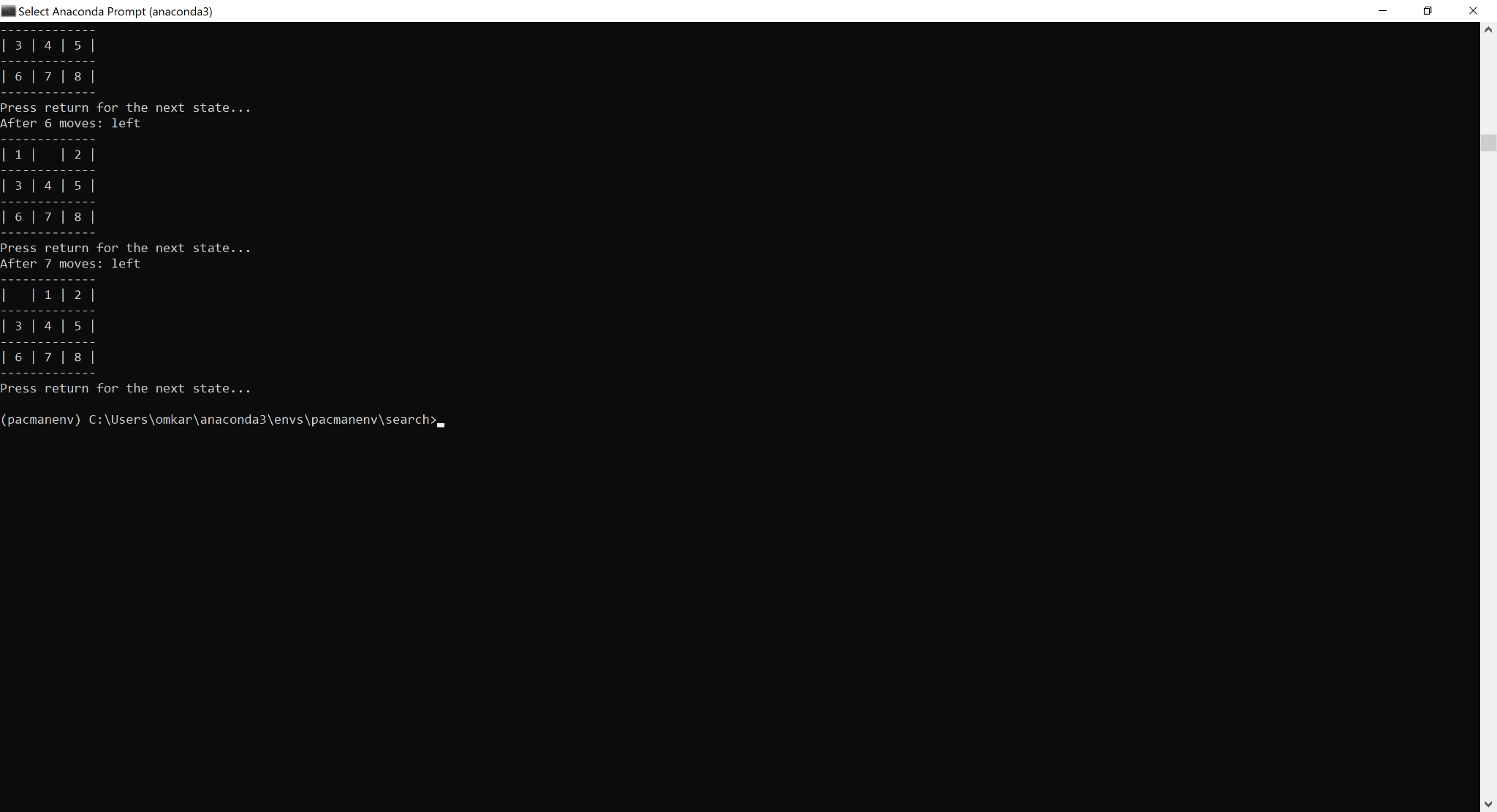
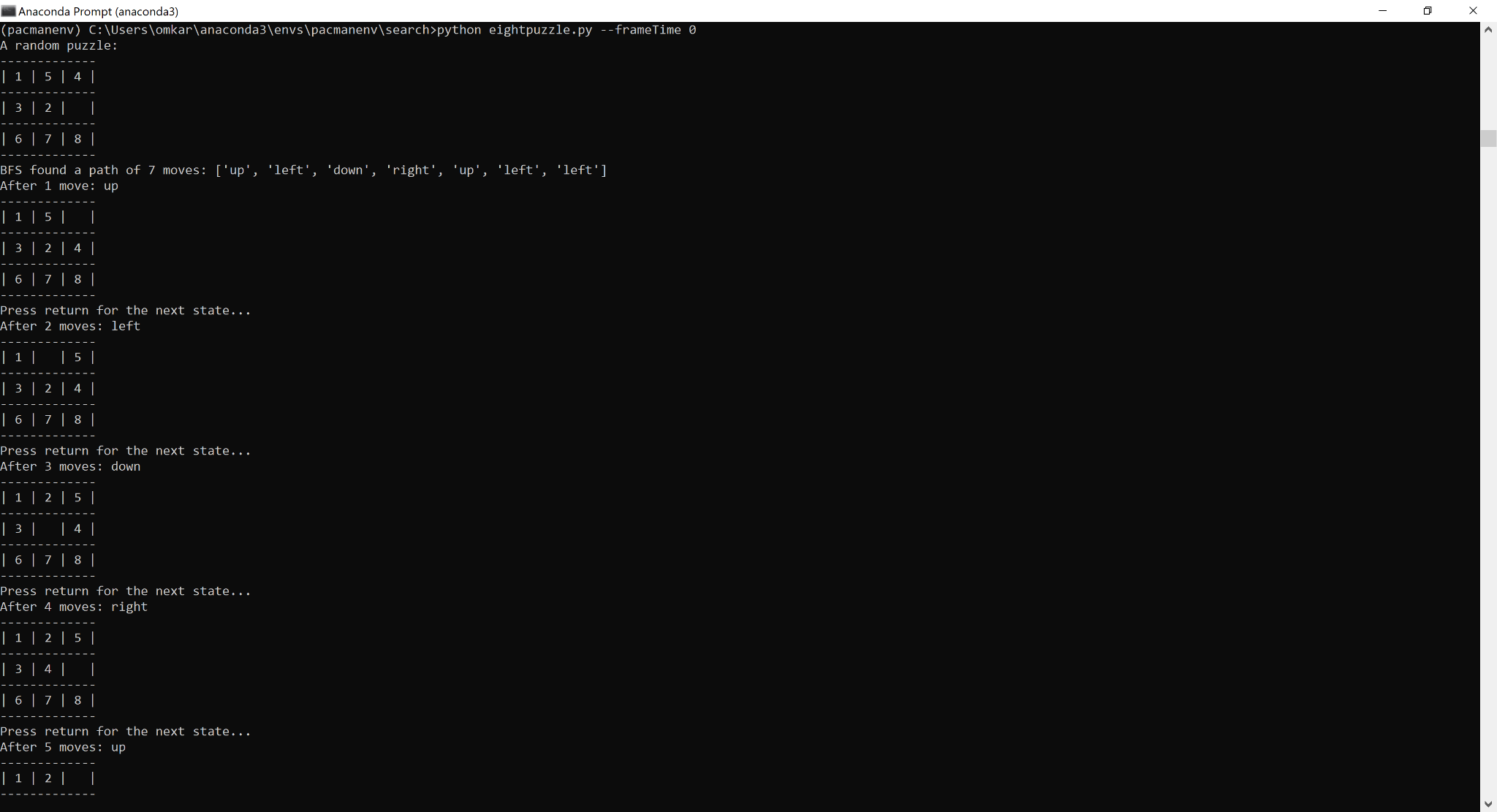
Q: Is this a least-cost solution? If not, think about what depth-first search is doing wrong.

A: This isn’t a least-cost solution because DFS explores the deepest node before backtracking and stops if a solution is found and hence is not optimal.

**Question 2:**



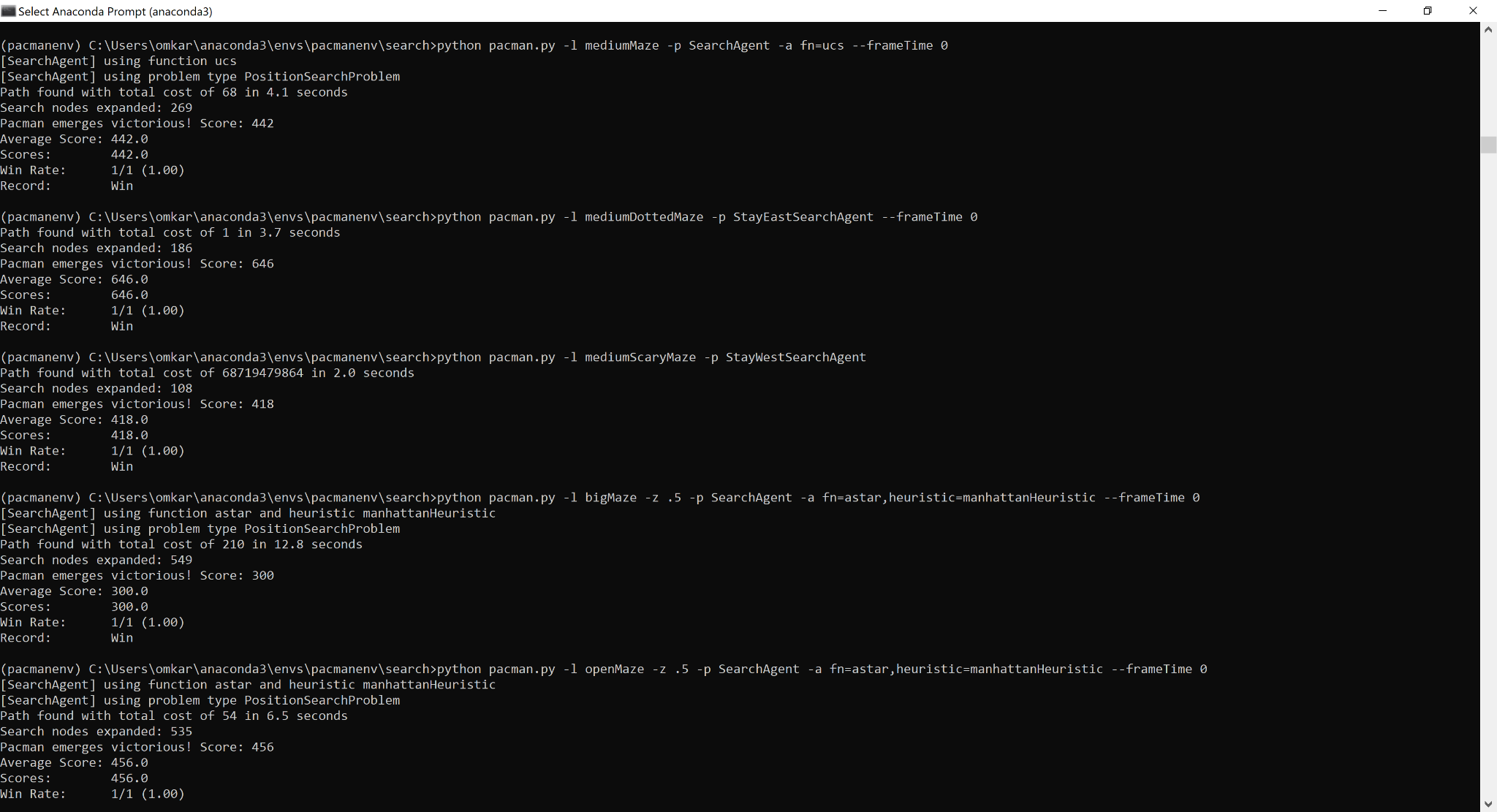
Eight puzzle:



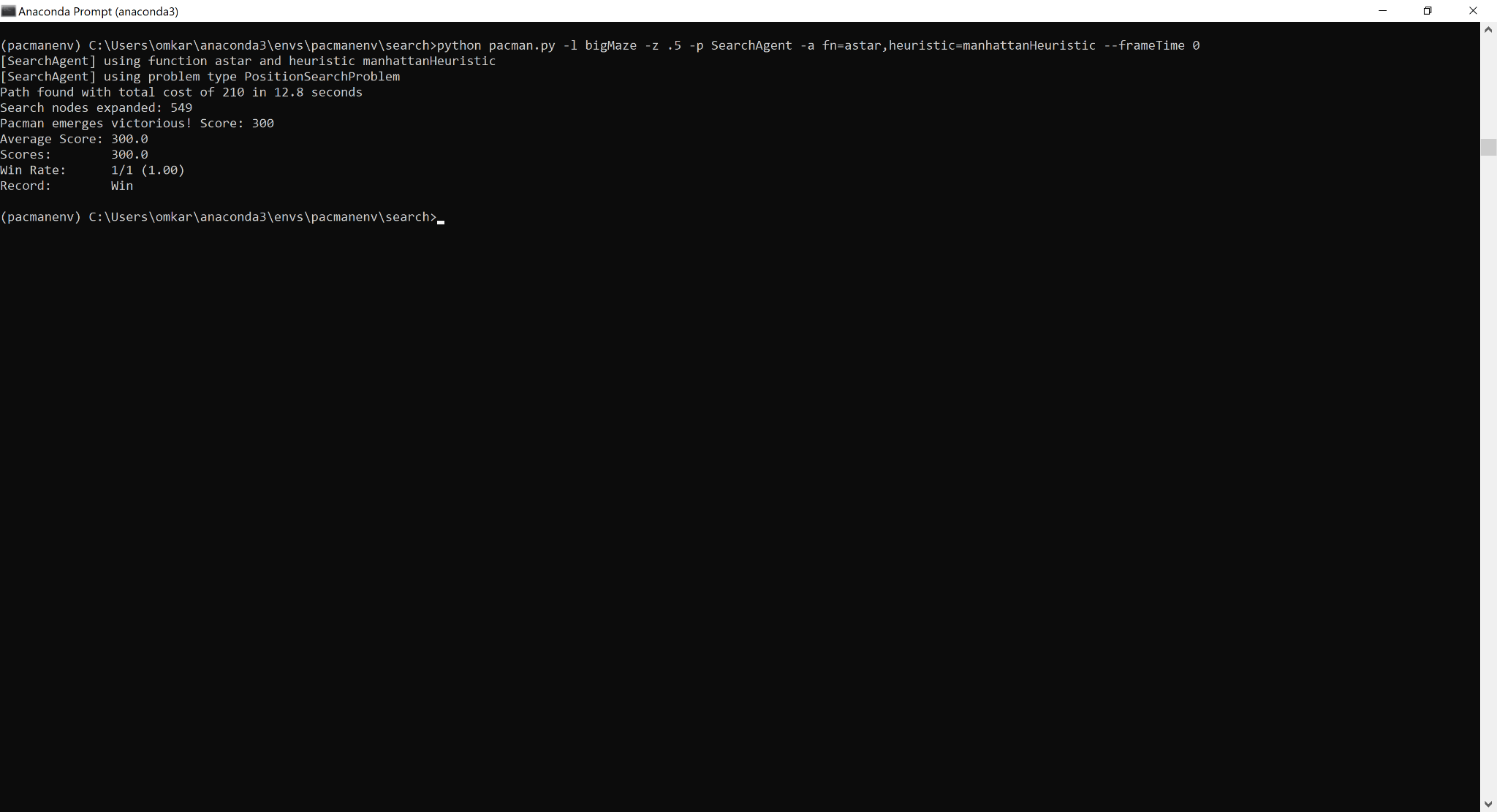
Q: Does BFS find a least cost solution?

A: Yes, BFS finds a least cost solution as it explores the shallowest node first hence all the nodes shallower than the goal node must not have satisfied the goal condition thereby ensuring optimal solution provided all paths are of equal length.

**Question 3:**

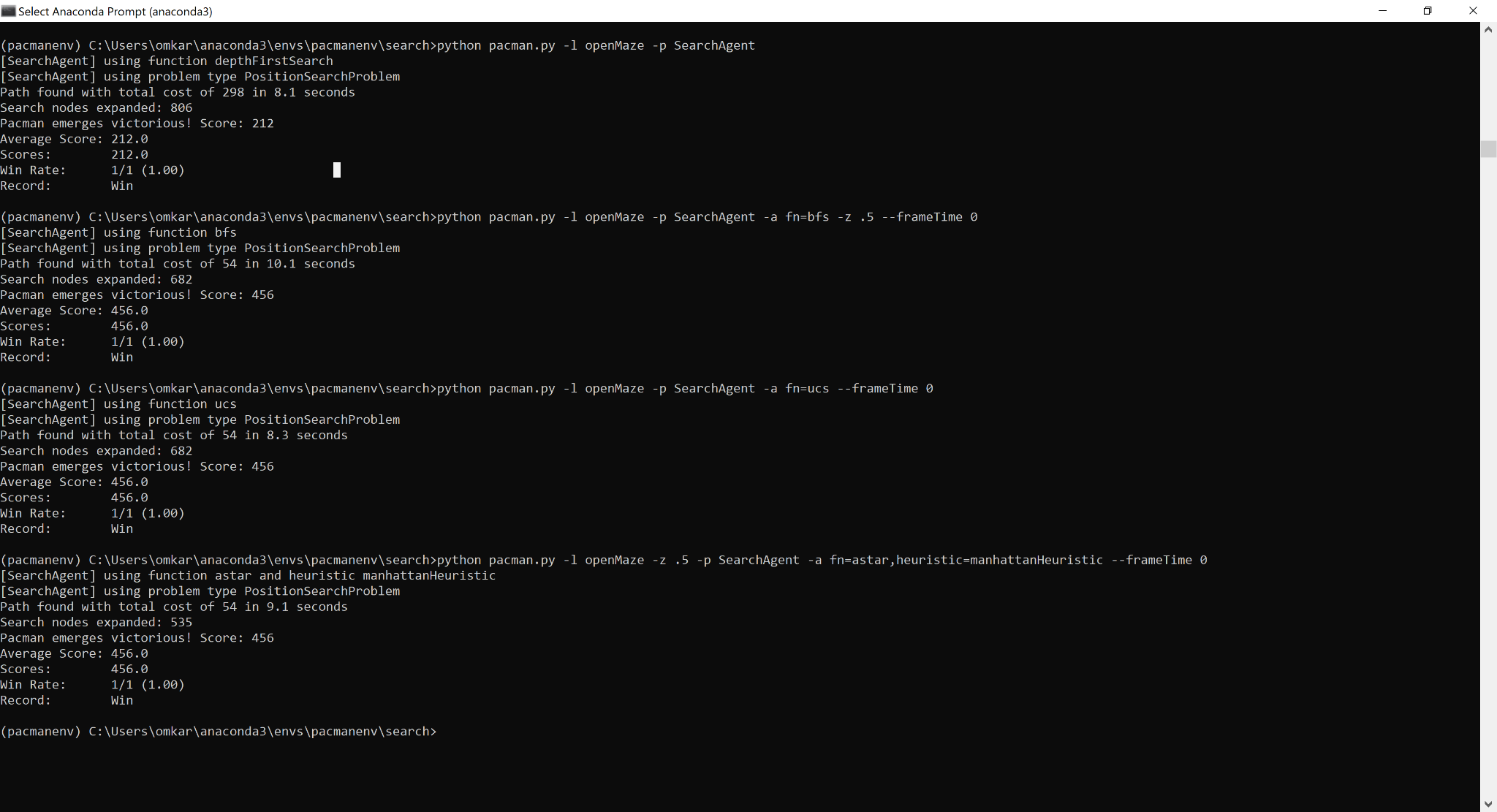


**Question 4:**

****

Q: What happens on openMaze for the various search strategies?

A: In open mazes there are multiple solutions possible and since DFS doesn’t look for the most optimal solution it performs poorly compared to other strategies.



**Final Autograder:**