Week 5

Progress:

We’ve been successful in solving the first six questions of the problem statement. The first 4 being finding the path to a certain position using different algorithms namely DFS, BFS, Uniform Cost search and A\*. The next 2 questions were to find the shortest path to travel all four corners of the maze using BFS and A\*. We’ve successfully accomplished solving both of them.

Learnings:

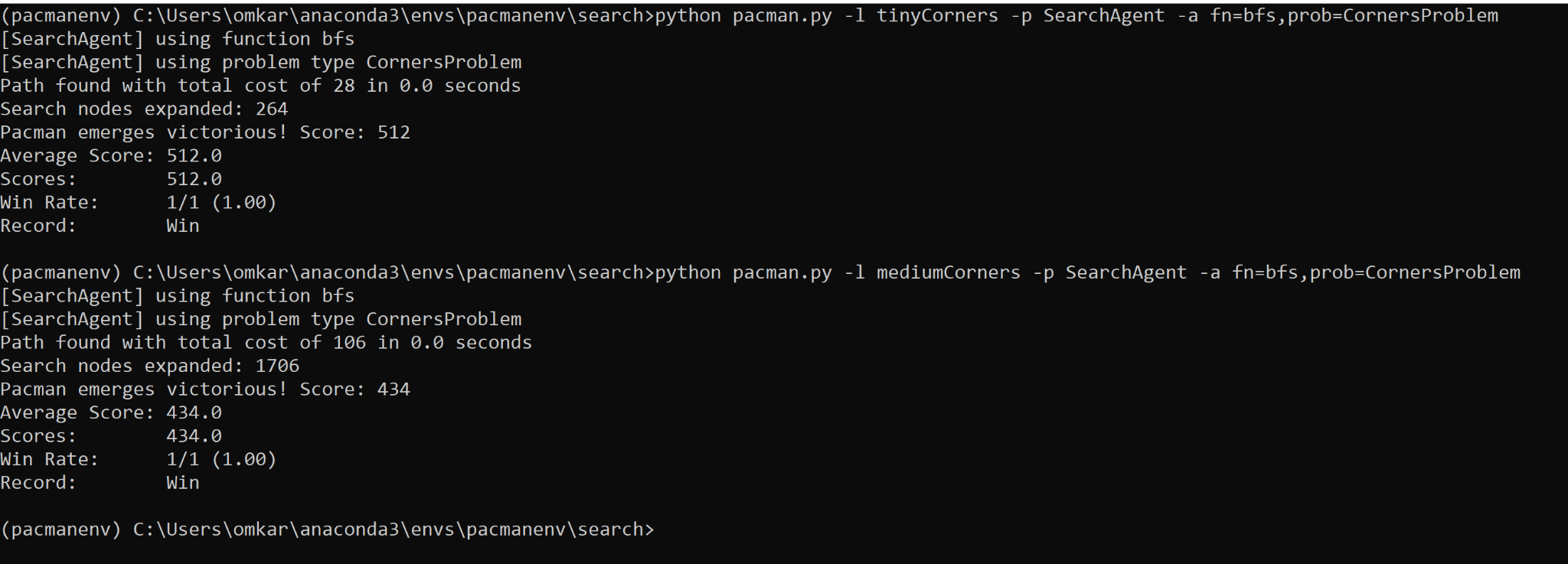
We initially only checked whether all four corners are being visited in BFS without putting any constraint on the first corner to be visited but observed that it always explored the closest corner first and the final path obtained was not optimal. We learn that visiting the closest corner is not an optimal way to tackle this problem, in fact for the tinyCorner problem visiting the furthest corner first gives the most optimal path. We tackled this problem by finding all possible paths and then choosing the most optimal one. Whereas for A\* no such extra effort was required it automatically chose the path with optimal overall cost. The major learning of the week was to learn to define the game state which enables the Pacman to make optimal decisions in cases of multiple goal states like when there is food in multiple places.

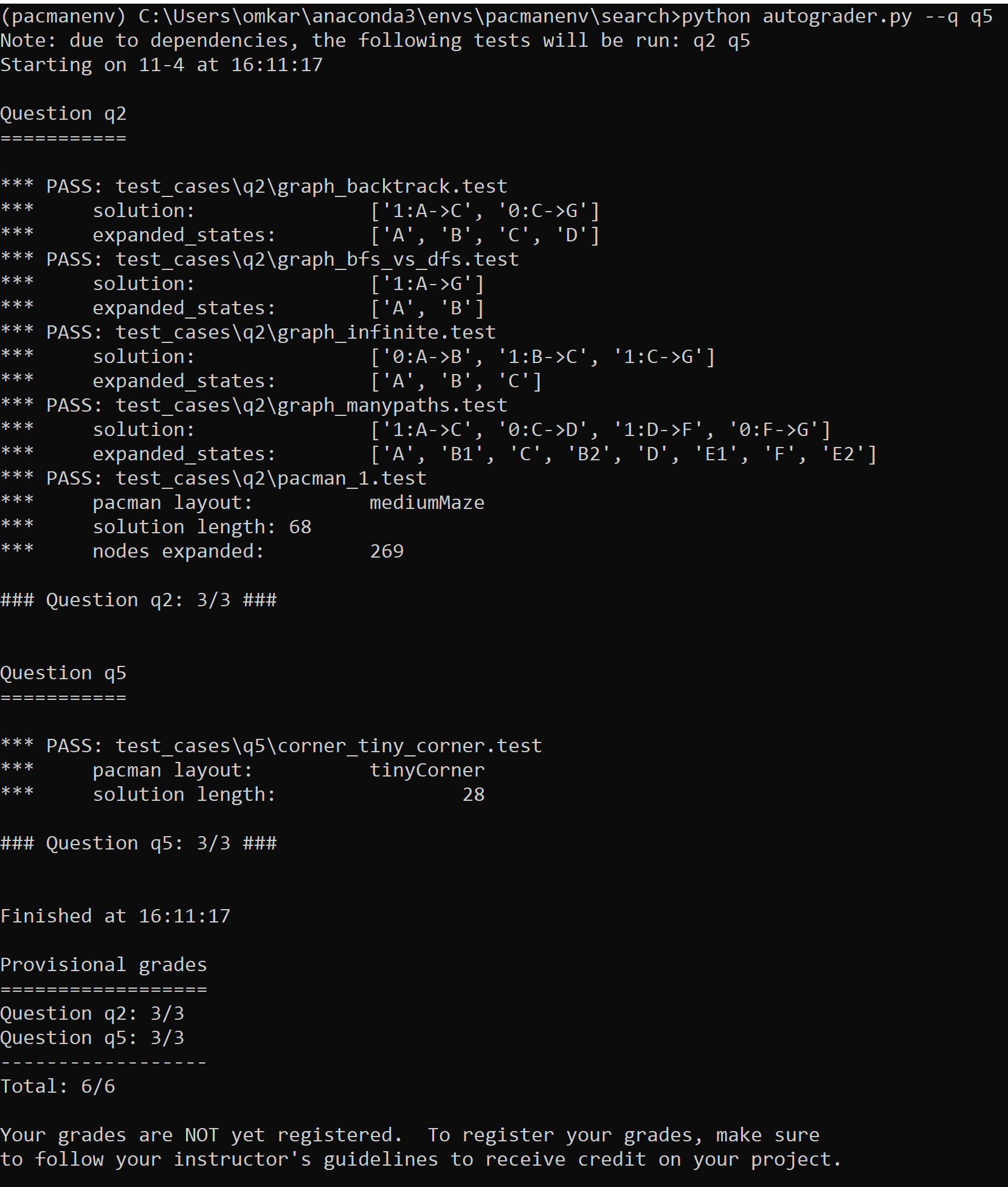
File Listing and Description:

search.py: It contains all the algorithms used by searchAgents to find path through the maze. In this week we modified functions for the search algorithms, BFS and A\* to accommodate a more complex goal state to deal with the CornersProblem.

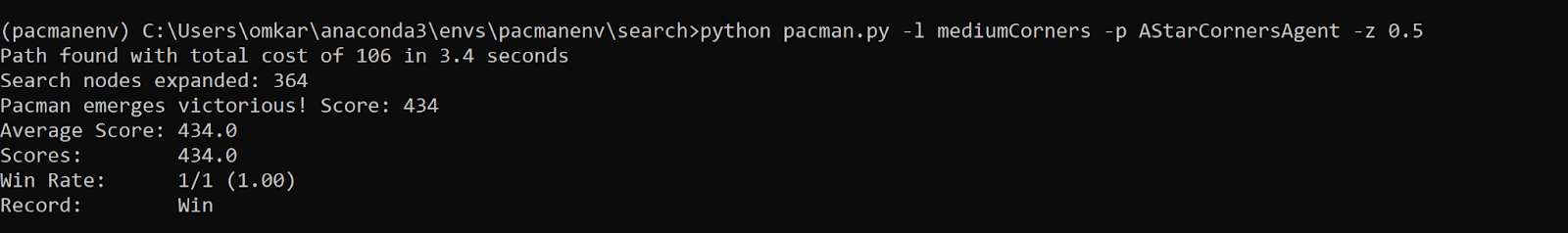
searchAgents.py: This file contains all the agents to run pacman and the different problem states are defined here. We modified the CornersProblem class to define start state, goal state and heuristic function.

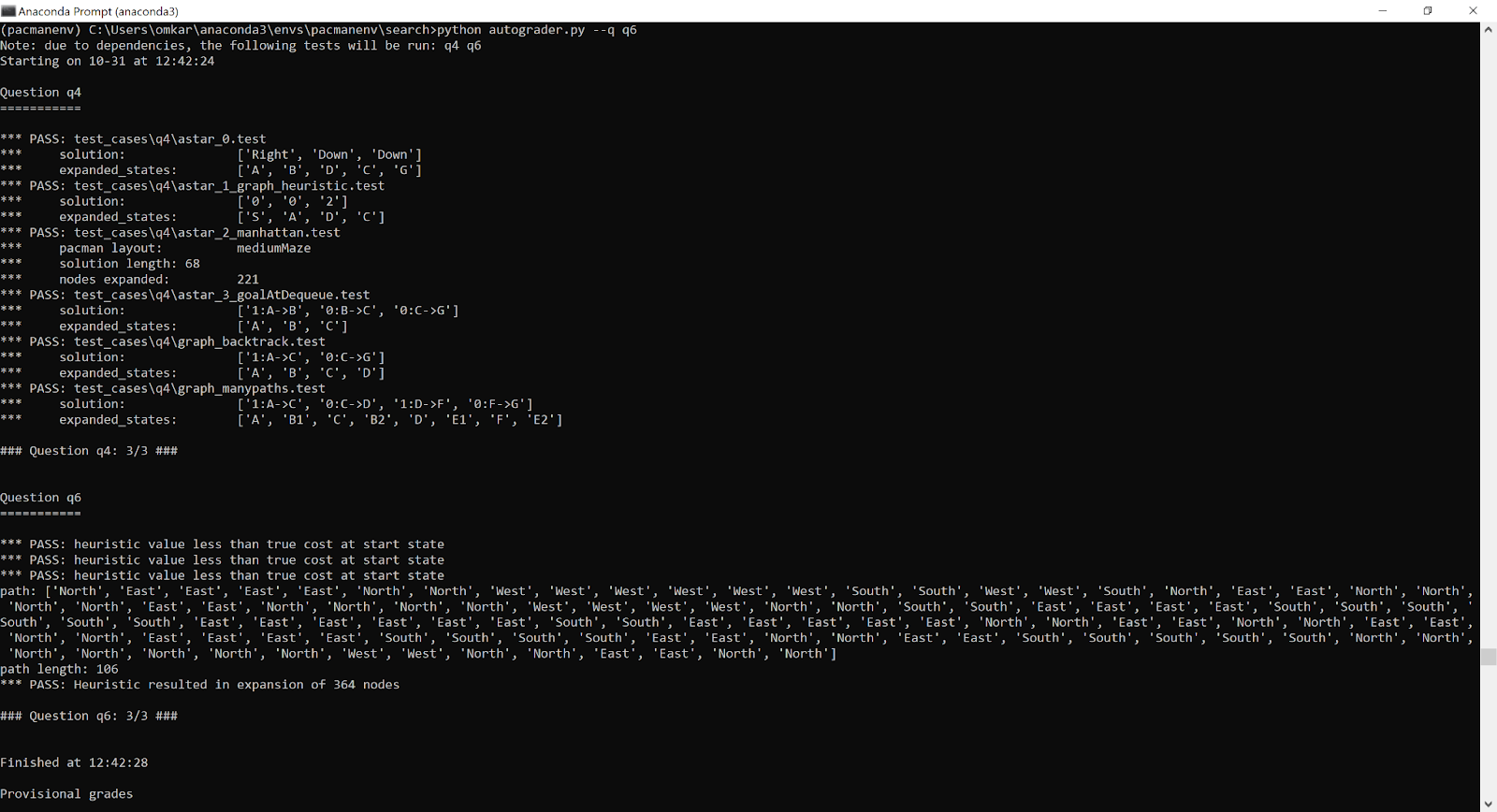
**Question 5:**

****

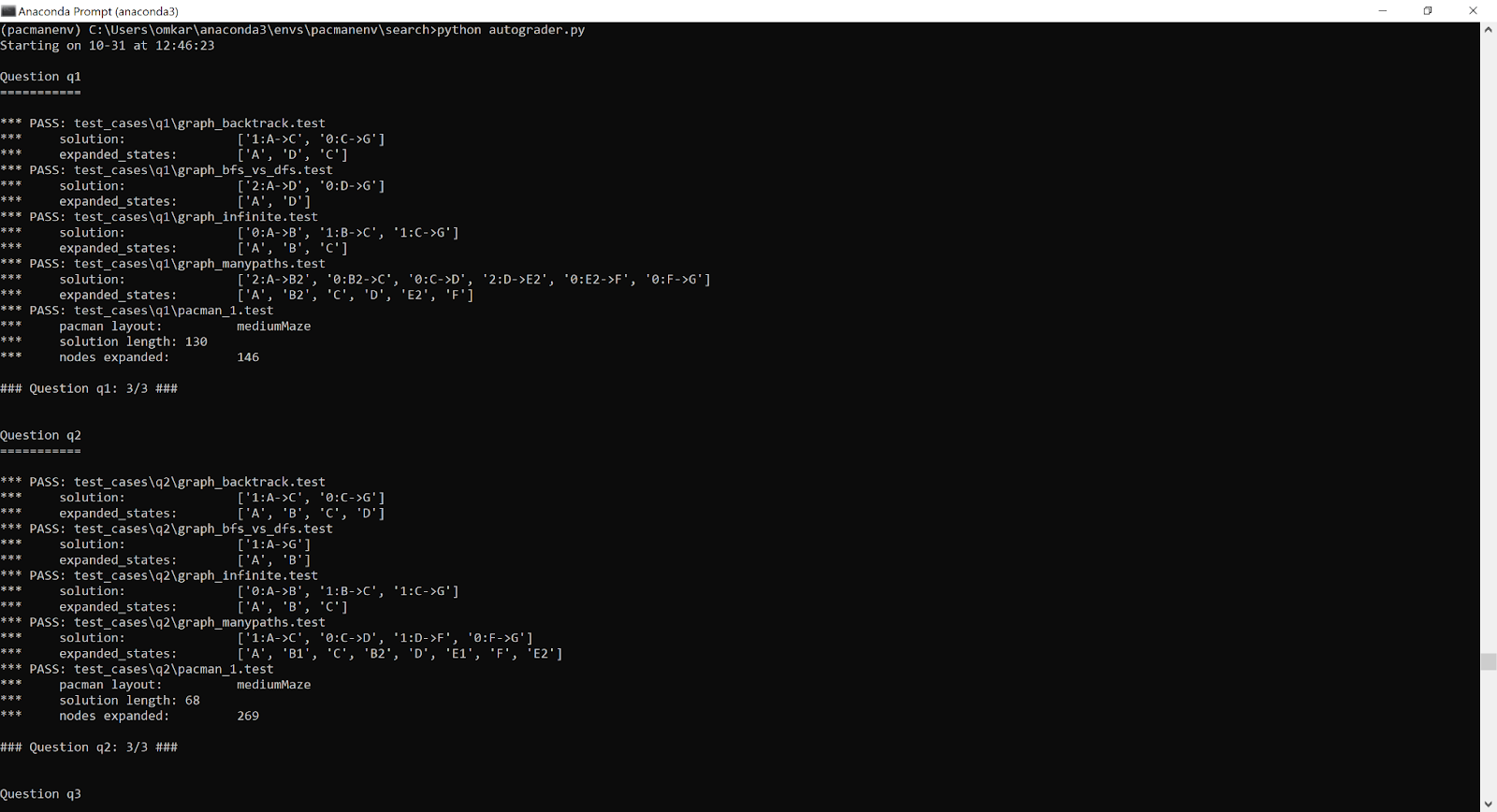
****

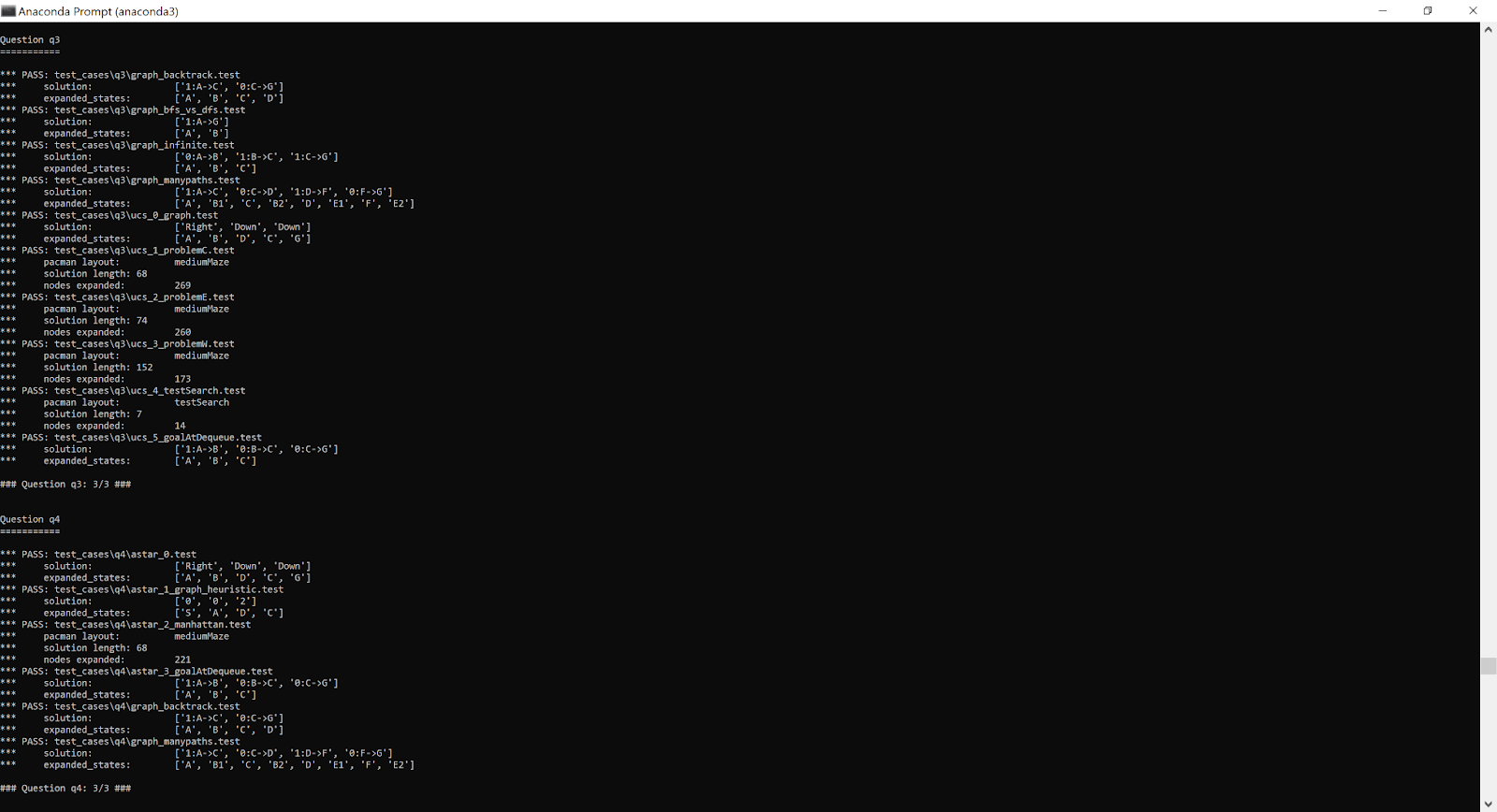
**Question 6:**

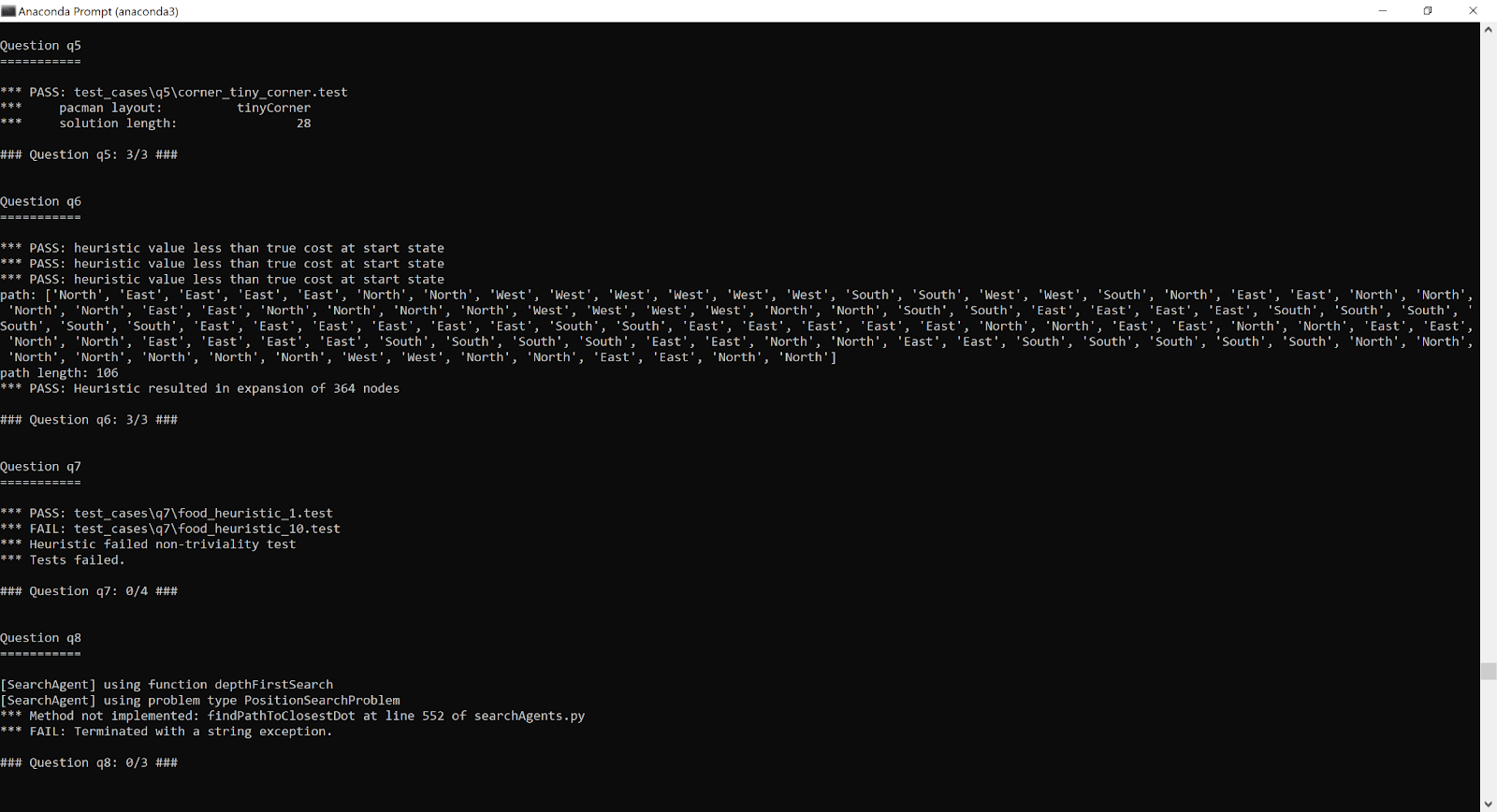
****

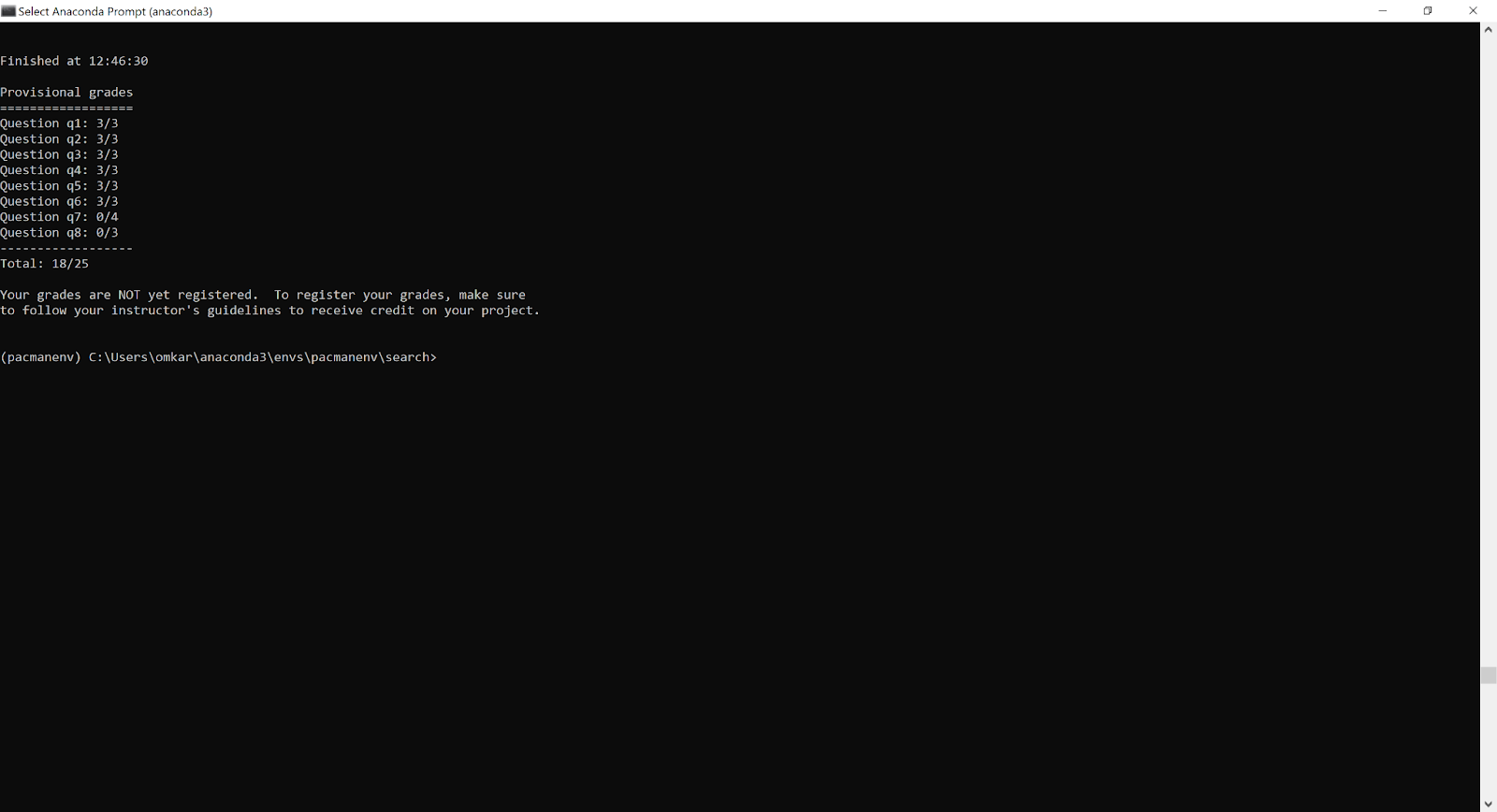
****

**Final Autograder:**

****

****

****

****