**CS205 - PacMan Week 6 README**

[Gayatri Bhosale](mailto:bhosale.gayatri@gmail.com) : 862394365

[Sanjana Senthilkumar](mailto:sanjana.senthilkumar@email.ucr.edu): 862384460

Puneet Singhania : 862327375

**Question 5: Finding All the Corners:**

We implemented the CornersProblem search problem in searchAgents.py where we chose a state representation that encodes all the information necessary to detect whether all four corners have been reached. From the screenshots below (after the autograder screenshot), it can be seen that our search agent solves tinyCorners and mediumCorners both. The state representation does not encode irrelevant information like the position of ghosts and extra foods.

* For tinyCorners: The cost is 28 which is the shortest path as suggested on the website.
* For mediumCorners: The cost is 106.

getSuccessors() generates successor states while it is on its path and captures the triples (successor, action, stepcost).

**AUTOGRADER RESULTS:**

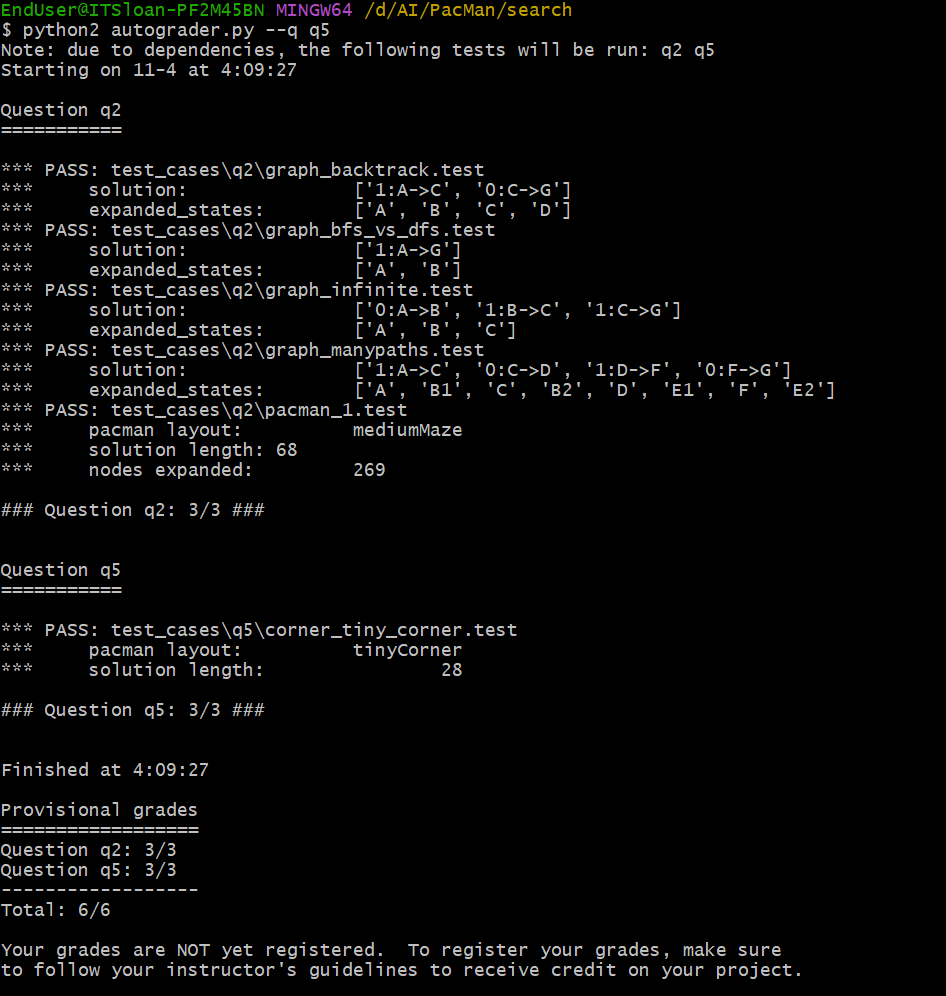


Figure 1: Autograder for Q5

**SOLUTION FOR THE PACMAN COMMANDS PROVIDED:**

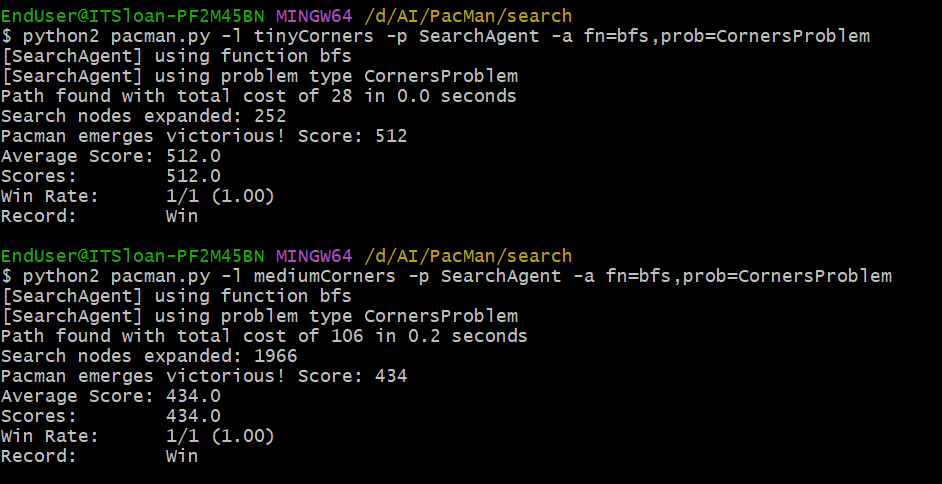
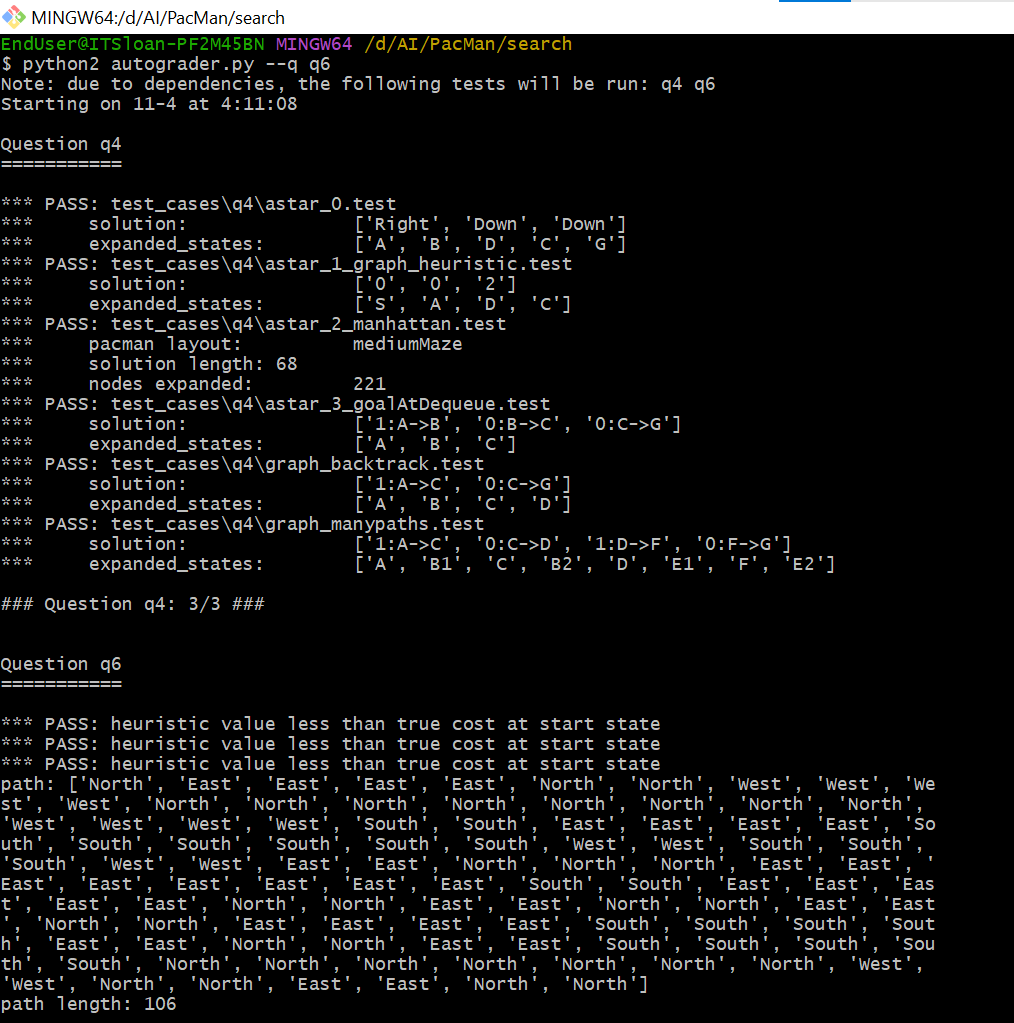


Figure 2: PacMan commands for Question 5

**Question 6: Corners Problem: Heuristic:**

In cornersHeuristic(), we have implemented a non-trivial heuristic that satisfies admissibility and consistency. Here we computed distances from the current position to all other unvisited corners. We are using Manhattan distance which is better than Euclidean distance to find the nearest corner. Our algorithm yields a value of 0 when a goal state is reached, and it guarantees the absence of negative values in its output. From the screenshots below, it can be seen that only 783 nodes were expanded.

**AUTOGRADER RESULTS:**



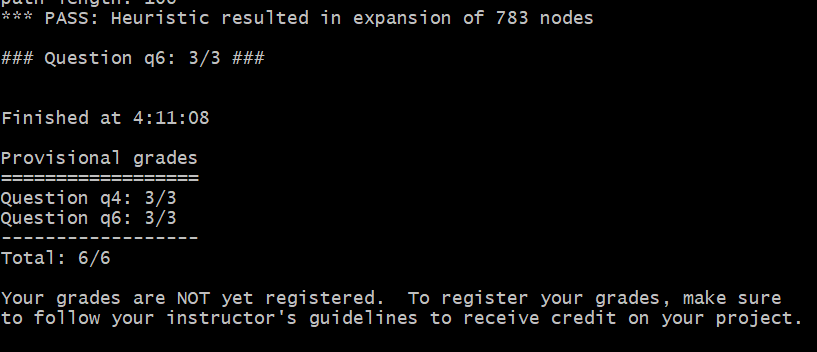


Figure 3: Autograder results for Q6

**SOLUTION FOR THE COMMANDS PROVIDED:**

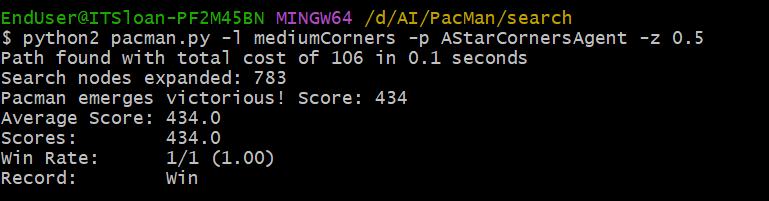
****

Figure 4: PacMan commands for Question 6

**Learnings:**

This week we learned about how Pacman can touch all the corners. This implementation was challenging but we could define a state representation that does not encode irrelevant information like the position of ghosts and the position of extra food. The initial code with Pacman GameState as the search state was too slow and we had to try many times to get the optimized start state. This taught us that the initial state is important too! We also learned about heuristic functions which can be trivial and non-trivial and how they affect the compute time in the algorithm.