Week 4 README - CS205 PacMan Project

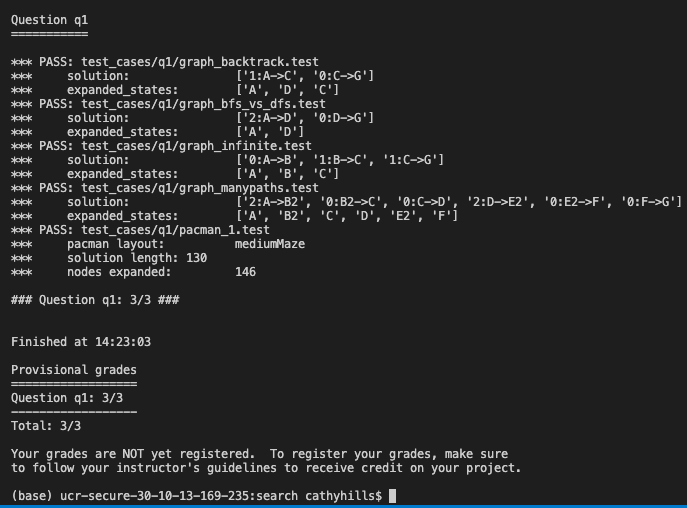
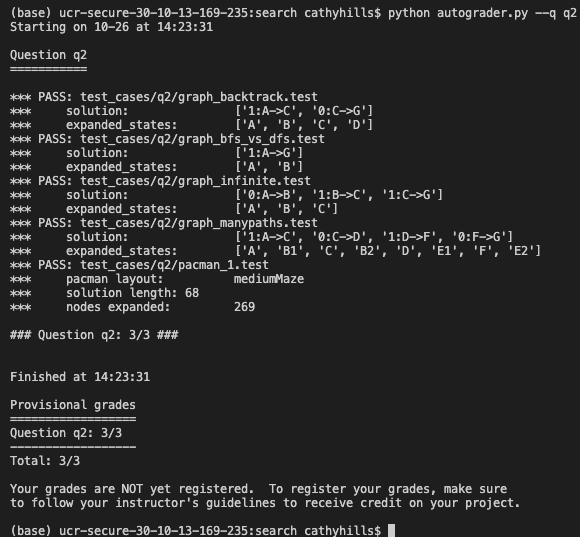
Anushka Tiwari, Bryce Hills, Rajat M. Jain

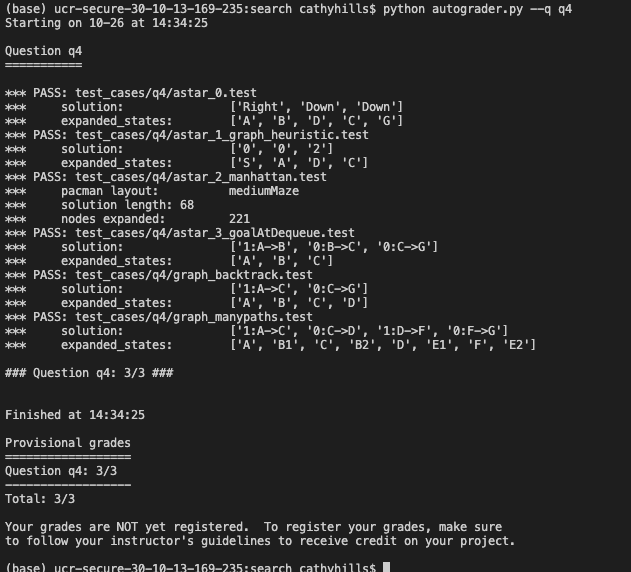
Week 4

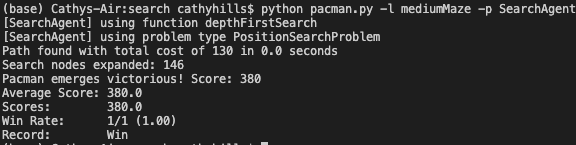
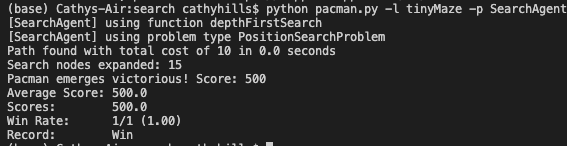
* **Summary of Learnings/Progress from Week 4:**
  + Progress: We have successfully implemented all four algorithms: DFS, BFS, UCS, and A-Star. We ran multiple test cases on each to verify that the algorithms are working correctly and to compare the costs between different games for each of the algorithms.
  + The main takeaway from this section of the project is that the order in which the states are added/removed from the data structure is important because it will affect the cost of the game. This is partly why UCS provides the least cost solution since it pushes/pops to the priority queue based on the cost value.
* **File Listing and Description:**
  + search.py
    - This file includes all of the algorithms which will be used to implement the searches into the pacman game. We implemented DFS, BFS, UCS, and A-Star algorithms in this Python file. Various other predefined methods are also included in this file which are mostly utilized within the search algorithms implemented by us.
* **Question 1**
  + **The Pacman board will show an overlay of the states explored, and the order in which they were explored (brighter red means earlier exploration).**
  + **Is the exploration order what you would have expected?** 
    - There are cases where pacman starts with only one possible direction and the nodes that correspond to that direction are explored first. This makes sense since DFS expands depth first.
    - Other cases where Pac-Man can move in multiple directions and it expands a specific direction first is also expected if we assume that direction is the first added to the actions.
    - Overall it is expected since DFS will explore nodes depth-wise once initialized with the start node.
  + **Does Pacman actually go to all the explored squares on his way to the goal?**
    - No, many explored nodes are not traversed on the way to the goal for all board sizes.
  + **Is this a least-cost solution? If not, think about what depth-first search is doing wrong.**
    - No, dfs does not take the cost of explored nodes into account in terms of adding to the data structure, thus the order in which nodes are added to the data structure could be more optimal.

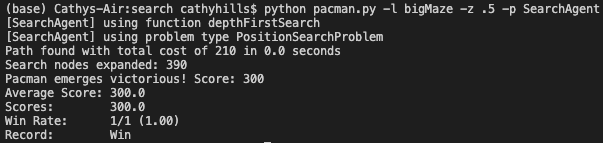
**Screenshot Section**

**Autograder Results:**

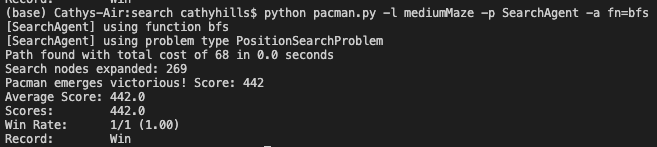
 

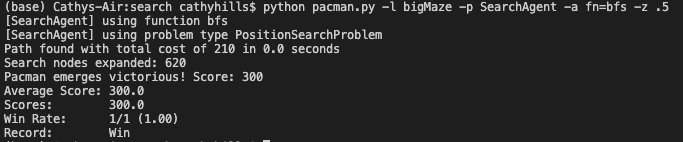


**Running DFS on Various Sizes:**

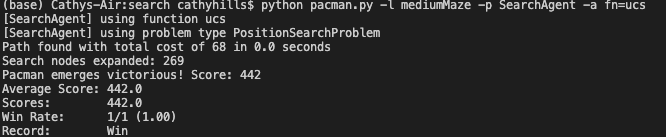


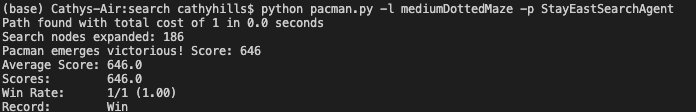
**Running BFS on Various Sizes:**





**Running UCS on Various Sizes:**





**Running A\* on Various Sizes:**

