Homework Assignment 2

600.335/435 Artificial Intelligence
Fall 2015
Due: October 15th

Game Playing

In the past weeks, we discussed intelligent agents and how they can use tree searching techniques to solve abstracted problems. In this assignment, modeled after the Berkeley AI course's search assignment, we will use a variety of searching techniques to find a route for pacman to take to reach the pellet. For this assignment you are allowed to work in pairs. Note that grad students will be expected to complete some additional tasks (if a pair of students has one undergrad student and one grad student, we expected all grad student-specific tasks to be completed).

First, download the code from the Berkeley website (search.zip). Follow the instructions in order to be sure your system can properly run python and start the game. Please take a look through search.py and searchAgents.py to see which functions you will need to implement, as well as pacman.py, game.py, and util.py. Please fulfill all of the following items and provide all performance-relevant output information (in case we aren't able to get your code to run properly):

- Complete question 1, the depth-first search. Provide your solution for tinyMaze, mediumMaze, and bigMaze.
- Complete question 2, the breadth-first search. Provide your solution for tinyMaze, mediumMaze, and bigMaze.
- **Grad Students**: In addition to the two previously-mentioned search algorithms, please implement an iterative deepening search. Provide your solution for tinyMaze, mediumMaze, and bigMaze.
- Complete question 3, which varies the cost function of your breadth-first search. Provide your solution for mediumMaze with the UCS agent, mediumDottedMaze with StayEastSearchAgent, and mediumScaryMaze with StayWestSearchAgent.
  Grad Students: Create a new cost function and provide the solution it generates with mediumMaze, mediumDottedMaze, and mediumScaryMaze. Why did you choose this cost function and how did it alter the results you see for these environments?
- Complete question 4, the A* search using manhattanHeuristic. Provide your solution for tinyMaze, mediumMaze, and bigMaze.

- Complete question 5, which solves the corners problem with a BFS agent. Provide your solution for tinyCorners and mediumCorners.

- **Grad Students**: Complete question 6, which creates a new heuristic for the corners problem. Provide your solution for tinyCorners and mediumCorners. How did you choose this heuristic and how well did it perform?

- Complete question 7, which solves the eating all the dots problem with A* with a null heuristic. Provide your solution for testSearch and trickySearch.
  Grad Students: Implement foodHeuristic and provide your solutions using this new heuristic for testSearch and trickySearch. How did you choose this heuristic and how well does it perform?

Submission Requirements

You should email Paul (`pwilken3@gmail.com`) a zip file containing:

- A README that provides all necessary instructions for running your program to verify the results you provide, as well as an explanation for any parts of the assignment that aren't functioning properly. If you are able to figure out the nature of the problem you're experiencing, this will be looked upon more favorably than a non-functioning program with no explanation. You also need to include the names of the students who worked on the assignment (limited to 2!) and a breakdown of the division of labor between the students. If your group has an undergrad student and a grad student, this section is important as it determines our grading policy for the undergrad student.

- The search.py and searchAgents.py files with all functions implemented as instructed. Please be sure to include informative comments!

- A text document (pdf or word format preferred) that answers any prompts in the above questions. Note that for this assignment the questions asked by the Berkeley professor can be thought of as simply thought exercises, but questions asked in this document should be explicitly answered.