Sam Traylor, Vaughn Bangash

Project 1 Written Portion

Collaboration Plan: Vaugn and I pair coded this lab together, not over zoom but in person as we live together. We coded on two separate computers but discussed the best ways to approach the problem when stuck rather than divvying up questions.

Question 1: The key difference between the DFS and BFS algorithms in pacman is the key difference in the way they work in general: the frontier is managed differently. The DFS algorithm uses a first-in-first out queue meaning the first element added would be the first one to be removed, and subsequent removals target the node added least recently. Conversely, BFS uses a last-in-first-out stack, where the topmost node will be the one to be removed. The functions are generalizable because they use robust data structures from util.py which allow any GameState (any valid pacman level) and SearchProblem to be navigated, and either search function can be selected at runtime for use in solving a particular level.

Question 2: In the StayEastSearchAgent, the cost function for stepping into a position (x,y) is 1/2^x, penalizing staying on the West side of the board. Conversely, the cost function for the stay east search agent has a step cost for entering position (x,y) of just 2^x, penalizing staying in the east side of the board. These encouraged behaviors are useful as the essentially allow designing an agent whose reflex is to favor one side of the level, and similar cost function manipulations can be used to achieve other goals in pac-man.