

Assignment 8

I (Chase Yates) and Liam Astin worked on this assignment. I submitted the code to Gradescope

Q1

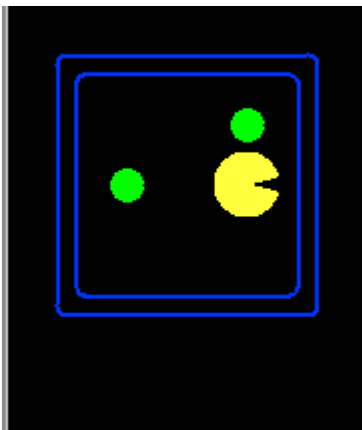
I was again the driver in pair programming for this week and kept the same partner. The experience was fine.

Q2

For the DFS, I used down, up, left, right as the ordering. It didn't find the closest goal and actually found the goal the furthest up. DFS traversed 85 nodes.

Q3

It is possible for DFS to find a path to a goal that is not the closest goal while searching fewer nodes than BFS. The conditions that would lead to this are when the current point in the search has two goals fairly close. The DFS must first search in a direction that leads directly or near directly to a goal before BFS has a time to search the neighbors of a cell, or must search the neighbors of other cells in its queue. One simple example in a 5×5 maze is this:

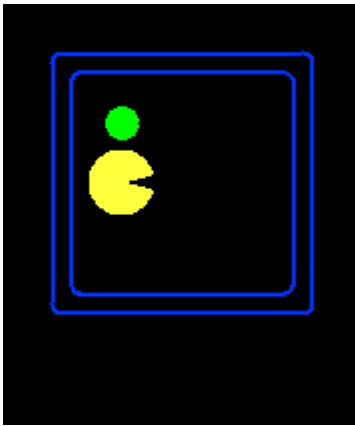


In this example, the top goal is the closest, however if DFS had its first neighbor checked be the left neighbor, and BFS searched the top last, then DFS would go left, left and find

the further goal in 2 checks, whereas BFS would check the left, bottom, right, top, twice as many checks before it found the goal.

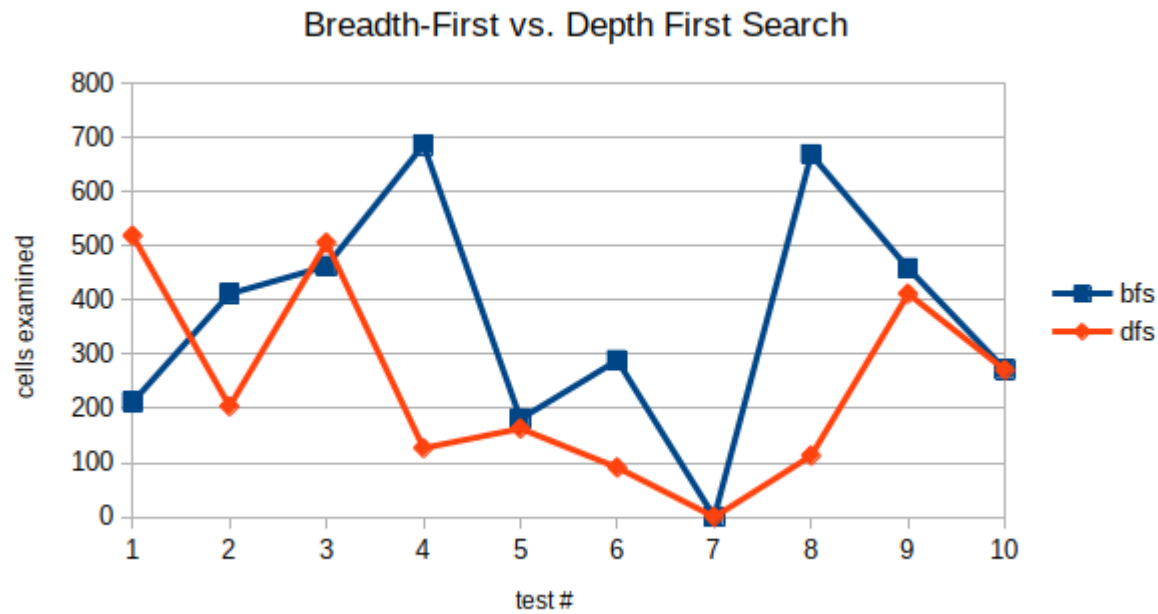
Q4

In the following example in a 5×5 'maze', given a neighbor search ordering of down, up, right, left, BFS would find the goal in $O(1)$ time, searching a maximum of 4 nodes in a different ordering, but only 2 in the order specified, while DFS will search the entire maze in a snake like pattern, going down, right and up, right and down, and so on, until the rightmost wall is found, and it will return, eventually returning next to the goal and finding it only then.



Q5

In my experiment I made 10 mazes of size 50×50 with 30% wall density and 5 goals and tested how many cells were examined by DFS and BFS to compare them. Here was the following results:



As you can see, most of the time (with some exceptions) DFS searched fewer cells than BFS. This is likely due to simply how the two algorithms work. DFS searches in a line-style fashion, branching off when a dead end is found, which seems fitting for mazes, while BFS searches in a radial fashion, examining all paths of equal length 'simultaneously'. Although BFS is guaranteed to find the shortest path to a goal. DFS Seems to find a path sooner somewhat often.