**Step 1:**

* Directions.png – utility function that gets the cardinal directions for the path (*Directions*)

Text

Description automatically generated with medium confidence

*Directions*

* DFS1.png – Finally made my recursion work correctly, but the solution being returned is not correct (*DFS1*)
  + Instead of returning the path from the goal node, I think the other search paths are being included in the solution as well
  + The plan is to add some print statements to see what the fringe and expanded set hold at each iteration of recursion to see what’s going on

Graphical user interface, text, application

Description automatically generated

*DFS1*

* DFS2.png - Decided to memoize the search and pass in a path parameter
  + The path parameter memorizes the path from child to parent
  + Keys are children and values are parents
  + Path starts from key (0, 0) in order to get the last movement of pacman before the goal
  + Then a while loop reconstructs the path to the goal starting from the goal and prepending the values of the path until we get to the start state
  + Figured out that this needed to be called recursively so state was added as a parameter in order to track the state currently being called in the recursion
  + Removed solution from the parameters, because it is now reconstructed after all the calls have been removed from the stack

Graphical user interface, text, application, Word

Description automatically generated

*DFS2*

* Directions2.png - added a second optional parameter (opposite) which lets you flip the value of the cardinal direction
  + Necessary when trying to find the last move pacman makes to get to the goal state

A picture containing table

Description automatically generated

*Directions2*

* Search.png – modularized the search algorithm so it work on all searches, and the only thing different is the fringe
  + Read over the notes and the idea of a generalized search that differed based on the fringe seemed much easier to me
  + Problems:
    - Still not working for bigMaze
    - Bfs does not work with medium maze either
    - At first, the problem with BFS was that because the start state was being added to the fringe but was passed in as the first state, it was calling search on all states twice
      * Since been fixed
      * I don’t know how to fix the current problem yet

Text

Description automatically generated

*Search*

* Constructpath – modularized the algorithm for constructing the path from goal to start
  + Problems:
    - Takes a long time for both BFS and DFS in bigMaze and BFS in mediumMaze, I might need a better way to construct the path

Text, letter

Description automatically generated

* Search2.png – changed search to keep a running path from parent to child to try and speed up the algorithm
  + Got rid of constructPath method
  + Now search returns the solution to the search
  + Problems:
    - Path variable is taking up too much space since the next path is constructed from the path of a previous state. Need to find a way to remove previous states from path without removing paths I might need in the future

A picture containing text

Description automatically generated

* UCS.png – implemented the same way as BFS but using a priority queue
  + Problems:
    - Does not work at all because something is off with my logic

Text

Description automatically generated

* A\*.png - implements A\* search with a priorityQueueWithFunction object
  + Problems:
    - I think there is a problem with the heuristic function
    - An error occurs anytime I try and push a new state onto the priority queue
    - Might also be a problem with how I’m calling the recursive function

Text

Description automatically generated

* Search3.png – changed search so now all search algorithms work for it
  + Had to check if fringe was empty at the end of the function, instead of the first if statement
  + Now, the algorithms don’t have a redundant starting state in the fringe
  + Also, the searches using a queue don’t empty the fringe before finding the goal

A picture containing graphical user interface

Description automatically generated

* Search4.png
  + UCS works because for some reason, the python compiler doesn’t recognize if a parameter is True, only if they’re false
  + Problems:
    - Astar still not working
      * Push() causing an error

Text

Description automatically generated with medium confidence

* Search5.png: Changed the parameters being passed into Search for A\* search
  + I forgot that the heuristic needs to be calculated for each recursive call, so I’m passing in the actual heuristic to Search
  + Problem:
    - Heuristic still isn’t working
      * The problem might be that the actual heuristic function hasn’t been defined, I may need to create a function that can be called to calculate the heuristic value

Text

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

* getHeuristic.png: Created function that calculates the cost of all currents steps and adds it with the mannhattan distance of the current state to the goal state
  + realized that the heuristic had not actually been defined
  + went through the notes and remembered that I needed to make an admissible heuristic
  + problems:
    - Need to pass extra information into heuristic in order to get the cost of all the current steps and the Manhattan distance of the currect state
      * I wanted to make the function global so other functions can use it if its required in future steps, but I may just need to define it in aStarSearch
* Search6.png and A\*2.png: finally got A\* to work
  + The problem was how I was using the heuristic function
  + Had to actually define how the heuristic was calculated
  + Then I had to figure out how to calculate the heuristic by just pushing the item in the queue
    - Decided to make the item a tuple with the path and the current state
    - Now I can still use A\* in the current general Search function
  + Problems:
    - Doesn’t work with bigMAze
      * Most likely because elements that have already been expanded are still being added to the fringe

**Step 2**