Matt Dumford
CS411 - HW2

Breadth first search:

After parsing the arguments into a game board, the program adds the board to a queue. It will then repeatedly pull boards out of the queue. For each board it pulls out it will determine if that board is solved. If it's not solved it will calculate the valid directions it can move and generate the boards for those moves. It checks to see if it has already seen these new board states, and if not, it adds the new boards to the queue. This is repeated until the solved board is found. The list of moves it took to get to the board is tracked within the board class so the program knows how to recreate each board from the original.

Iterative deepening depth first search:

This is very similar to the breadth first search approach. The main difference is that it uses a stack instead of a queue. The program has a variable to keep track of the depth that it will search to. Then, each time it starts the search, it will create a new stack and push the original board on the stack. It then repeats the steps explained for the breadth first search, popping the boards off of the stack instead of removing them from the front of the queue. Whenever the search reaches the depth that it is limited to and the stack is empty, it increments the allowed depth and starts the search again.