In this programming task, we'll come back to writing Python scripts to implement search algorithms. Note that, if you do your assignment in Python, we expect that you'll submit your Python scripts to enable us to test your program.

In Week 3, we learned about uninformed search algorithms (such as DFS, BFS and Uniform-Cost Search). We looked at a few examples, including the Romania path search problem and the N-Puzzle problem. We will take advantage of the aima-python code base to learn about the search algorithms and how they can be implemented. Let's adapt this code base to implement the programs for the Romania path finding problem and the 8-Puzzle problem.

Task 1: Can you extend these programs to let them take as input a file containing the country (e.g., Romania) that you want to search for paths? Or, for the 8-Puzzle problem, the input file will contain the initial & goal configurations.

Task 2: Can you extend the 8-Puzzle program for N-Puzzle problems? For NxM-Puzzle problems?

Extension:
Refer to the tutorial 3L and 5L jug scenario. Attempt to use a BFS implementation to arrive at the same solution we discussed during the tutorial. Remember that you will need an appropriate data structure to represent the jugs in each state, and a way of determining what actions are possible given a provided state.