COS30019 - Introduction to Artificial Intelligence
Tutorial Problems Week 3

Task 1: Consider this problem: We have one 3 litre jug, one 5 litre jug and an unlimited supply of water. The goal is to get exactly one litre of water into either jug. Either jug can be emptied or filled, or poured into the other.

For this problem give:
(a) An appropriate data structure for representing a state.
(b) The initial state
(c) The goal test
(d) A specification of the operators which includes the preconditions that must be satisfied before the operator can be used and the new state generated.
(e) Draw the full state space
(f) What is the solution to the problem?

Task 2: In the previous exercise, a representation for states and the full state space were developed. For the same problem, apply search strategies and note:
- The order in which nodes are created in memory
- The nodes that are not created in memory at all for the following search strategies:
  a) Breadth first search with no checking for duplicate states
  b) Breadth first search with checking for duplicate states
  c) Depth first search with no checking for duplicate states
  d) Depth first search with checking for duplicate states

Programming Task – Uninformed Search

There are a few uninformed search algorithms discussed throughout the semester, each utilising a data structure referred to as the *frontier*. The frontier in each search is not necessarily represented the same way, however its function remains consistent: to store nodes that have been discovered, but are yet to be explored.

For this lab, you will be programming a *Breadth First Search* (*BFS*) implementation. Consider what this means of the frontier: What data structure would be most suitable for representing it?

The arguments passed to the main function of your program include the file to read from, the city to start the search at, and the city to end the search at. Using your solution to the problem for lab 1 (a file parser that handles map data), read in the provided text file and use your BFS implementation to determine the path used to reach the destination from the starting city.

You should begin by making sure that you can successfully read the file with your week 1 code, and print out all relevant information to the console. When this is achieved, you can proceed to programming your BFS algorithm.

The goal is to have the sequence of cities traversed from origin to destination displayed in the console, from start to finish.

Extension:
Refer to the tutorial 3L and 5L jug scenario. Attempt to use a BFS implementation to arrive at the same solution we discussed as a class. 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.