Intelligent Agents

Practical 3

PART 1 – Search

These questions are designed to ensure that you understand the core concepts of search. You may need to look up some of the terms used. Please attempt the questions before the practical so that they can be discussed there.

1. Consider the vacuum cleaner world agent. Assume that the agent perceives whether a room that it is in is dirty or not without needing to perform an action, has no knowledge of other rooms and has the following actuators (Left, Right, Clean). The agent also perceives walls (which determine whether a Left or Right action would succeed). The aim of the agent is to ensure that the (static) world is clean.
    a. Create an instance of Vacuum world that has 3 rooms with the agent starting in the left-hand room. Generate the complete search graph for the agent acting in the world to a sufficient depth so that you feel comfortable with the process.
    b. What is the branching factor of the graph? What determines this?
    c. What is the depth of the graph? What determines this?
    d. Is the graph also a tree? Explain.
    e. Step through a Breadth-First search on this graph. What did you learn from this?
    f. Step through a Depth-First search on this graph. What did you learn from this?
2. Consider the graph in Figure 1. Assume that you are starting from A and trying to reach G. You may assume that your search will not return to the last not it just left (*i.e.*, no A -> B -> A loops).
    a. Redraw the graph so that "left" means the number earlier in the alphabet, so the first three nodes should be read with A at the root, B below and on the left and C below and on the right. Nodes that are the same number of "steps" from the root (A) should be shown on the same level.
    b. Step through a Breadth-First search on the graph from a).
    c. Step through a Depth-First search on the graph from a).
    d. Repeat this but assume that you're using Dynamic Programming so that you will not repeat a search state.
    e. Repeat c) but use iterative deepening.
    f. Step through a Breadth-First search on the graph from Figure 1.
    g. Consider the effectiveness of these approaches on this graph.
3. Go to visualgo.net and experiment with these searches.

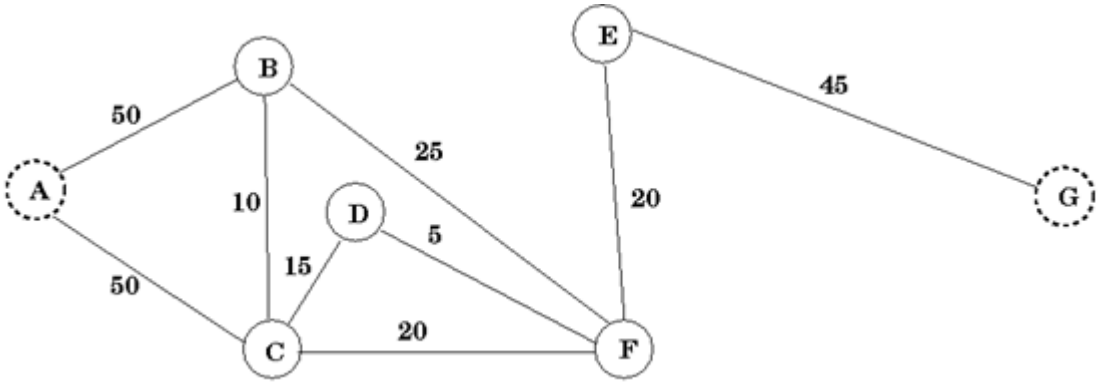

Figure 1: Graph for Q2, taken from COMP3006 material

PART 2 – Coding

We are going to experiment with the Pac Man library from UCal Berkeley. We are going to follow some of what is listed here.

These programs do not currently work on the lab computers due to a missing library, so I haven't put a question around this (I'll update this sheet once the library is installed). I've included the link above so that you can play with this on your own computer if you wish. Note that this is in Python, so if you haven't used that before have a look at the code and get familiar with it.

PART 3 – Coding Study

Part of the assignment is going to be implementing search algorithms. That means that you need to be able to ingest a search graph to run the algorithms on (which means loading it and storing it in an appropriate structure). Your task for this week is to code up this process.

The format of the graph file will be that each line of the file gives the details of one edge of the graph. Each line has three positive integers, separated by spaces. The first and second integers represent vertices that form the endpoint of the edge and the third is the "cost" of the edge.

For example, the first three lines of the graph in Figure 1 would be:

0 1 50
0 2 50
1 2 10

Note that I've equated A with 0, B with 1 and C with 2. You will need to load in lines until the file is completed. Think about a useful structure to store those edges in.

There is no restriction on what languages that you use and what libraries you use except that the result must work in the lab computers in 314.219. If you use code from another source, please use appropriate code. referencing.