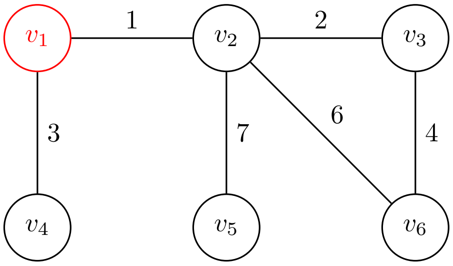
5COM2003: Artificial Intelligence

Worksheet 4
Mark Scheme

General Marking Instructions:

- Please provide reasoning for your marking decisions in the form of full sentences.
- Be polite in giving feedback.
- If you do not understand something or are unsure what something does, please state it.
- You should be able to run the code they provided yourself, so you should be able to check if the result is correct.
- Continuation errors only count once. (Sometimes this is tricky, yes.)
- If in doubt state your uncertainty.
- If you really can not figure out what is happening and you can not grade the work send me an email f.riegler@herts.ac.uk and we see how to proceed. (I expect you put in some work before doing this.)

This worksheet we are working on graphs, something you hopefully remember :).
We will go through all the steps to make a small agent "build" our spanning tree in accordance to Prim's algorithm for this. You can find plenty of pointers in the graph lecture.



(The numbers here are weights not identifiers.)

We start with v_1 as our first node.
From here we will need to select the lowest edge, here 1 to v_2.
The next step is to select between the values {2,3,6,7} we select 2 to v_3 as it is the lowest while not violating tree properties.
Next is v_4, followed by v_6. This leaves us with {6 and 7} as edge values, but the 6 edge would lead to v_6 and thus violate the tree properties.
We have to select v_5 via the 7 edge and we finish.

Building our system parts

1. (2 marks) Construct the graph given above. Give each node its name and each edge its weight.

1M If all the nodes are there and (mostly - ignore one or two mistakes) connected correctly
1M for the weights being (mostly) correct and stored on the edges

2. (3 marks) Write an agent object that can store the nodes and edges that are part of our current spanning tree. Expose these values through functions.
   Furthermore, give the agent the ability to sense its current node's name and all edges leaving this node (we are in a undirected graph).

   1M for an agent with lists of nodes and edges (one list, two list, does not matter)
   1M for the functions exposing this data
   1M for the sensing functionality

3. (2 marks) Write a function that chooses the edge with the lowest value that does preserve the tree properties. Assume a particular node is given. (The current agent's state.)
   This function may belong to the agent or the world. Either is fine here as the decisions of the agent will be honoured.

   1M For choosing the lowest weight edge
   1M For choosing the lowest weight edge that does create a circuit (violates tree properties)

Expanding the system

4. (2 marks) Add memory to the agent. Use this to store prior discoveries of nodes/edges, so the agent has a list of candidates to be added to our spanning tree (think back to AD about appropriate data structures to store available edge values). Add functions to expand and get information from this data structure.
   You may use Python inbuilt data structures for this.

   1M for the choice in data structure if it potentially solves the problem, i.e. a sorted list, a priority list, a heap, etc
   1M if it does so correctly

5. (3 marks) This final task might be a bit difficult. The goal here is to have a little agent that walks the current spanning tree in the making and expands it.
   The agent has senses and memory as described above, but may only add an edge+node once the agent is discovering them (from the adjacent node). As written in the function in task 3.
   The agent may only walk around the current spanning graph, using nodes or edges not in it is not allowed.
   The movement part of this is hard, an easier but inefficient solution to this is to use *randomWalk*, compare lecture 5, to get too the desired node.

   1M if the agent gets to the correct next node
   1M if the agent does so using only the current spanning tree
   1M for putting all the functions above together

Have a wonderful day :)