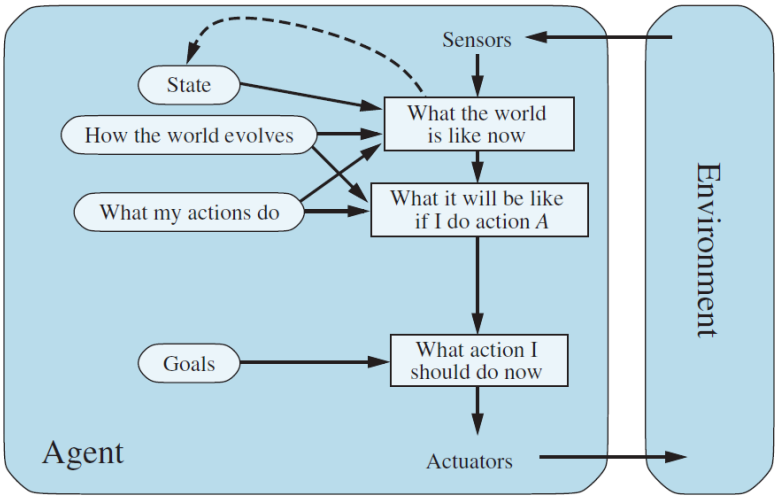
CSC4444: Artificial Intelligence
Spring – 2024
Assignment - 1 (10 Points)

Logistics:
Deadline: 02/08/2023 11:59 PM CT
Deliverable: A single PDF document containing the answers to all questions and a link to your program implementation.
Submission: Gradescope

Question 1 (3 points): Consider the following **goal-based agent** architecture:



Write the pseudocode for this agent, given the following construct:

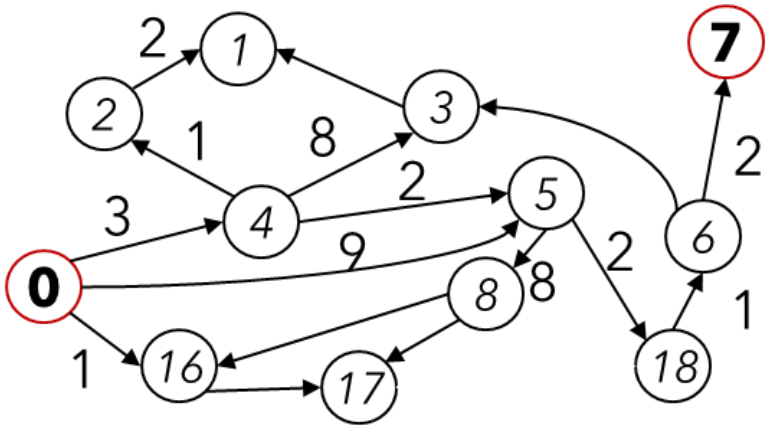
function GOAL-BASED-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    model, a description of how the next state depends on current state and action
    goal, a description of the desired goal state
    plan, a sequence of actions to take, initially empty
    action, the most recent action, initially none

Question 2 (3 points): Suppose two friends live in different cities in the same country. On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city *i* to its neighboring city *j* is equal to the road distance *d(i, j)* between the cities, but on each turn, the friend that arrives first must wait until the other one arrives before the next turn can begin. We want the two friends to meet as quickly as possible. Formulate this situation as a search problem. Use formal notations if you want. You must define the following:

a) State space
b) Initial state
c) Goal state/goal test
d) Transition model/successor function
e) Action cost function

Question 3 (4 points): Implement the uniform cost search using Python. We will use Jupyter Notebook for this implementation. The instation and startup guide for Jupyter Notebook can be found here - https://noteable.io/jupyter-notebook/install-jupyter-notebook/

Once you install Jupyter Notebook, download the base code as an ipython notebook - uniform_cost_search.ipynb. **You do not have to change the base code**. There are places marked on the notebook, where you have to add your code. Essentially, you will have to implement the ***uniform_cost_search*** function. The input to this function is the start state, one or more goal states, the input graph and the cost of edges in the graph. Your implementation of the ***uniform_cost_search*** should return the minimum cost(s) of reaching the goal state(s) from the start state. As an example, for the following graph, the minimum cost of reaching from node 0 to node 7 is 10.



This graph is already inserted as an example input in the notebook to be used as a unit test. However, this may not be the only graph we will use for testing. **Your code should provide a generalized solution**. Feel free to test your code with additional graphs as input by following the example input graph.

Once complete, upload the modified notebook to your favorite repository (Google Drive, One Drive, GitHub) and post the link as an answer to question 3 in the PDF.

Bonus Question (2 points): In addition to the minimum cost(s) in question 3, return and print the path with the minimum cost(s) from start to goal state(s).