Lab 2

Objectives
- Python - graphs
- Search

Important: Labs are to be completed individually. Do not collaborate with anyone in completion of this assignment. Do not post solutions, code, or these questions online, or make them accessible outside of this course.

I.   The Game of Nodes

Task

Winter is coming, and a war is on. You will help two sides in their missions: the White Walkers and Jon Snow.

Details

Your python program will consist of three files: one to handle a graph of nodes and two to handle two search agents. The search agents will be independent - that is, only one agent will run at a time. Both agents will use the graph of nodes, which is a map of cities.

Provided with this lab description is **data.csv**, which contains node coordinates in the following format:

> Winterfell,10,10

where the node for Winterfell is located at (x,y) coordinate (10,10). Additionally, the file contains node mappings in the following format:

> Winterfell,White Harbor,65

where an edge exists between nodes Winterfell and White Harbor, and the weight on that edge is 65. *The above two examples are format examples only.*

game-of-nodes.py
> This python file will handle the logic for the graph, including loading of graph data from the file. Running this file should display a map (graph) of the given data, with labeled nodes and edges.

white-walker-search.py
> The goal of the White Walkers in their search is to reach every city on the map (..generally bringing death & destruction, but we aren't worried about that part). They begin at The Wall and move together (they should be considered as one

agent). Running this search program should print the sequence of cities to the optimal solution.

jon-snow-search.py
>   The goal of Jon Snow in his search is to reach The Wall, as efficiently as possible. He begins in Trader Town. Running this search program should print the sequence of cities to the optimal solution.

Rubric

(10 points) Requirements - zip file and python files properly named, submission adheres to requirements.
(10 points) Comments - All .py files have a header comment containing your full name and abc123. Code in all files is easy to read and is commented for clarity.
(20 points) Game of Nodes Map - provide a screenshot of the map your program produces in the Lab2-Questions.pdf.
(20 points) White walker search
(20 points) Jon Snow search
(5 points extra credit) The results of the White walker search and Jon Snow search agents (sequence from start to solution) are also displayed on the graph of nodes.

II. Analysis

Submit your answers to the following questions in the file named Lab2-Questions.pdf. Your full name and UTSA ID must appear at the top of the page.

For complexity questions, if more than one algorithm applies, explain each in your answer. Assume the following notations (consistent with the textbook & lecture):
> b = the maximum branching factor of a tree
> d = the depth of the least-cost solution
> m = the maximum depth of the state space
> C^* = the cost of the optimal solution
> l = the depth limit of the tree

1. (5 points) Which searching algorithm is optimal for the white-walker-search implemented in question I? (Name which algorithm you implemented) Explain your answer.

2. (5 points) What is the time complexity white-walker-search implemented in question I? Show your work and express the answer in Big O notation.

3. (5 points) Which searching algorithm is optimal for the jon-snow-search implemented in question I? (Name which algorithm you implemented) Explain your answer.

4. (5 points) What is the time complexity jon-snow-search implemented in question I? Show your work and express the answer in Big O notation.

III. Bonus Question

If the search algorithms for Jon Snow and the White Walkers were run simultaneously, in which city would they meet? *Add your answer to Lab2-Questions.pdf*

Submission

On Blackboard, submit a zip file named **abc123-lab2.zip** containing only:
- Lab2-Questions.pdf
- game-of-nodes.py
- white-walker-search.py
- jon-snow-search.py

(where *abc123* is your UTSA ID)