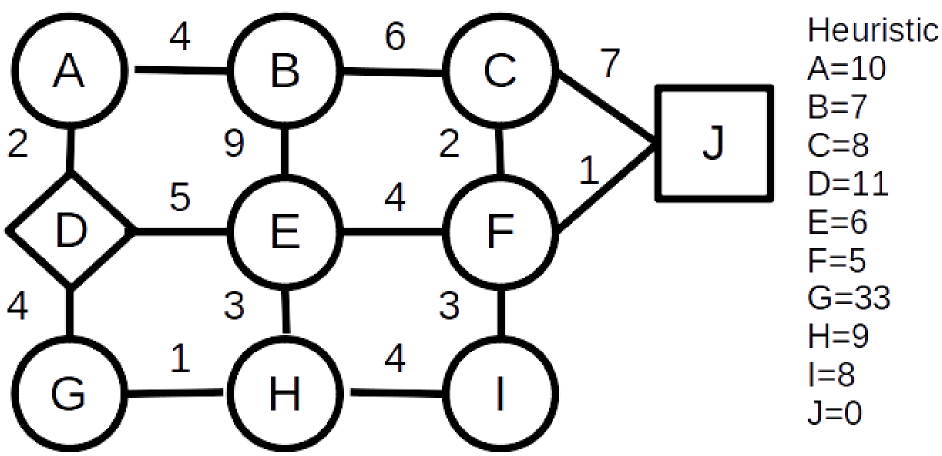
CSci 4511
Final

Name: ________________________________

Student ID: ______________________

Instructions: The time limit is 75 minutes. Please write your answers in the space below. If you need more space, write on the back of the paper. The exam is open book and notes. You may use electronic devices to ONLY look at either an e-book version or electronic notes. You may not use the internet, program/run code or any other outside resources. (If you are typing on your keyboard/input device for anything other than ctrl-F to find words in the e-book or notes, this is probably not acceptable.) For all questions you must **show work**.

Problem (1) [15 points]
Run the A* algorithm (graph version) on the following graph to find the shortest path. You start at "D" and the goal is "J". Is this heuristic admissible? Is it consistent?



Problem (2) [20 points]
Your uncle asks you to help out in his grocery store over the summer. You decide you want to rearrange where the items are located in the store to try and get more money. Answer the following:

- Classify the environment based on the seven criteria we discussed in class (fully/partially observable, single/multi-agent, deterministic/stochastic, episodic/sequential, static/dynamic, discrete/continuous, known/unknown). Give one sentence supporting each classification.
- Describe how you would represent a state. Then give a sample state, the possible actions from that state and the resultant states.
- Is this approach incremental or complete-state?

Problem (3) [20 points]
Give an example of a graph with a single goal where Uniform-cost search, A* search and hill-climbing search can be run with the following condition. Uniform-cost search will find the goal faster than A* and A* search will find the goal faster than hill-climbing (both of these conidtions need to be true on the same graph). Here "faster" means the number of nodes each algorithm investigates, not the solution quality. Do not have any "ties" that can be aribtrarily broken when running the algorithms that would impact the "faster" condition.

Problem (4) [20 points]
For each of the following problems, choose which algorithm from uninformed, informed and local search is best suitable to solve the problem. Support your choice a few sentences explaining what about this algorithm makes it superior. You may assume you have a heuristic for each of these problems.
(1) You are planning a car trip from Minneapolis to a city on the west coast. You want to get there as fast as possible.
(2) You are hired to help plan advertising for a product. There are a lot of options on how to advertise. (How much for online ads? Which sites? How many TV ads? How much to spend on production for TV ads? etc.) The company expects an outline of a advertising proposal in a few days. The goal is to make the most profit on the good being advertised (minus the advertising budget).
(3) You want to solve a large n-queens problem, such as 25 queens on a 25 by 25 board. (Note: This problem has a large amount of possible "goal" states.)

Problem (5) [25 points]
Assume I want to start making seating assignments for this class (a fixed number of rows and a certain amount of seats per row). You may assume there are more seats than people. Everyone submits preferences on who they want to sit next to (and who they do not want to sit next to). Sitting directly next to someone (left or right) gets their full happiness/preference, but sitting 2 seats away (left or right) gets only 50% the preference. People more than 2 seats away, does not impact a person's happiness. No preference is gained from the seating arrangement of front/back (to simplify problem). A person's happiness is the weighted sum described above of the preferences of the 2 people to the left and right. My goal is to maximize the average happiness due to seating. Give a heuristic for this problem by:

- Describe how you would represent a "state" when you model this problem.
- Describe how you are going to relax the problem. (This should not be a trivial relaxation.)
- Describe how you would optimally solve the relaxed problem without search to compute a value.