CSci 4511
Midterm 1

Name: ____

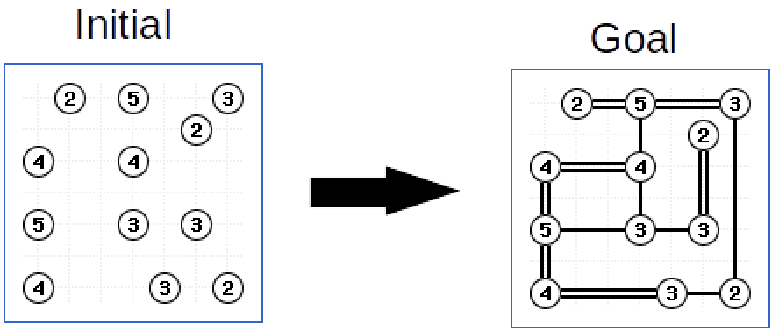
Student ID: ______________________________

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) [20 points] Suppose you have a computer that can store one million nodes/states in memory. Your problem has a branching factor of 10 at every node/state. In relation to the depth of the problem, state when you should use breadth-first search versus iterative-deepening depth-first search.

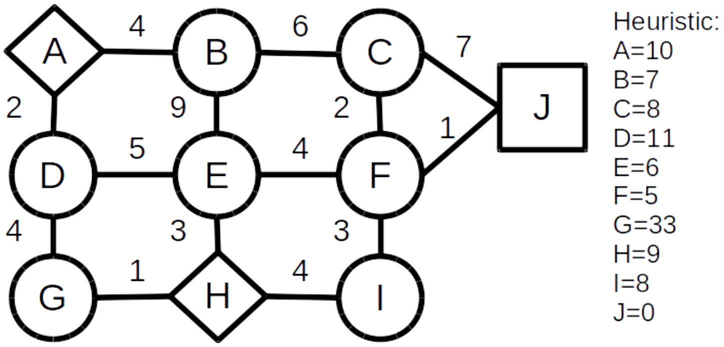
Problem (2) [20 points] Below is a picture of the game “Bridges”. It is played on a graph where bridges can only be connected in the cardinal directions (i.e. North, South, East and West). Each node must have a number of bridges connected to another node by the number represented in the center of each node. However, at most 2 bridges can be connected per pair of nodes and bridges cannot cross each other on the graph.
For this problem:

- Give a concise description of what states and actions look like for this problem
- What is the maximum branching factor of your approach?
- What is the maximum depth of your approach?
- Is the approach incremental or complete-state?
- Use your answers above to give a (reasonable) upper bound on the runtime and memory if depth-first search is used.



Problem (3) [20 points] This problem continues off the “Bridges” problem described in problem 2. Provide **two different** (non-trivial) relaxations for this problem. For each relaxation, describe how you would find the optimal solution for the relaxed problem. Which relaxation do you think is better? Justify your answer.

Problem (4) [20 points] Perform local beam search with 2 beams on the graph below. The initial points are indicated by diamonds (Nodes A and H) and the goal is a square (Node J). Show enough work step-by-step so that we know you are actually running local beam search.



Problem (5) [20 points] Make a graph where A* expands more nodes than uniform-cost search to find the optimal solution. Note, your graph needs at least two paths from the initial state to the goal (i.e. really a graph and not a tree in disguise). After you make this graph, answer the following questions with either a graph which has the property or a sound justification for why not:

- Is it possible to make a graph with an admissible heuristic that will cause A* to be slower than uniform-cost search?
- Is it possible with a consistent heuristic?