CSci 4511
Midterm 1

Name: _____

Student ID: _____

Instructions: The time limit is 75 minutes. Please write your answers in the space below. 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 search the internetor use other outside resources. For all questions you must **show work**.

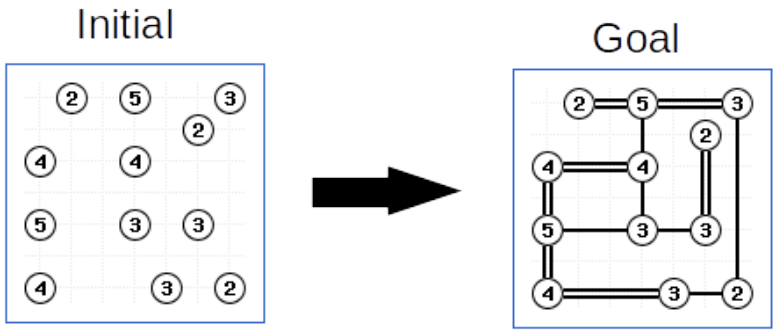
Problem (1) [20 points]
For each of the seven environment classifications except for known/unknown, indicate which is the more difficult one to handle. Then give an example of a "real world problem" (i.e. something with a name) that fits this classification.

Problem (2) [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 (3) [20 points]
Assume you have a problem with an admissible heuristic $h(n)$ for an A^* search. This problem could also be done with a bi-directional using breadth first search as a sub-search. Find a formula to estimate which search would be more efficient on this problem. (Note: Assume BFS's big-O bound is a fair approximation of the actual runtime.)

Problem (4) [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 (5) [20 points]
Make a graph where A* expands more nodes than uniform-cost search to find the optimal solution (without depending on any tie breaks causing the difference). 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?