Artificial Intelligence (Spring 2020)
Lab1 (Partner lab)

Due 2/6/20 (Thu) at 10:00pm via Box (See the last section for more information.).

Topics: informed search

Grading:
- **Part 1:** 60%
- **Part 2:** 30%
- **Reflections:** 10% (Refer to the last section.)

Write a program that solves the Traffic Jam puzzle. The puzzle consists of a number of vehicles on a grid parking lot. Each car has the size of two grid units, and each truck has the size of three grid units. Vehicles can move backward or forward in whole grid units as long as they do not overlap with other vehicles or go off the grid. You must implement A* to find the shortest sequence of moves allowing the red car to occupy the cell with the door on one of its edges (i.e., the red car can leave the parking lot). Note, moving a car multiple units in one direction is still considered a single move.

Initial State A

Initial State B

Initial State C

1. Report

Answering the following questions in 1 to 2 pages:

1. State space [You may assume that the door is on the top border of the parking lot, and the red car is aligned with the door.]
2. Initial state
3. Actions**:** Describe the general case, and specify edge cases.
4. Transition Model
5. Goal test
6. Lower and upper bound on the branching factor of this formulation, if computable
7. Lower and upper bound on the solution depth of this formulation, if computable
8. Optimal solutions to the 3 initial states provided above. For each state, specify the number of moves in the optimal solution, as well as the sequence of states in the optimal solution. Each state should be represented textually according to your state representation. There's no need for fancy graphics, but try to make it so that it is easy to see where the vehicles are located.

9. Performance Analysis: Implement A* with at least 2 heuristics (Explain why each heuristic is admissible). Compare the number of states explored/goal-tested until an optimal solution is found. Reason about the similarities and differences observed.

2. Source code

Provide a *briefly documented* source code in any programming language of your choice with a *readme.txt* file explaining how to compile and run the program. Implement everything from scratch, but you may use existing implementations of basic data structures (e.g., priority queue) and i/o functionalities. (You don't have to go overboard with commenting, but it should be easy to understand the code.)

3. Wrap up

3.1 Submission.

Steps:

- Open up a browser, and go to https://richmond.account.box.com/login
- Log in using your UR login.
- Navigate to a directory named after you netid. (It should have "Jon Park" as the owner)
- Click on [Upload] in the upper right-hand corner of the browser.
- Upload a directory named "lab1" containing the following files:
  - `lab1report.pdf (typed or scanned; Try to keep the file size smaller than 1mb by making it black and white, etc. Frankly, it could be under 100kb. Also, be sure **not** to include your name in the assignment for anonymized peer-assessment.)`
  - `A subdirectory containing the source code (the directory structure is up to you.)`
  - `readme.txt`

3.2 Meta-cognition. Now that you have completed the lab, it's time to reflect on your learning experience; please submit a short paragraph here: https://forms.gle/pxB22R2RQw2koh2JA. Remember, this is for you to better retain what you learned. The more you reflect, the more you will benefit from it. Reflect on anything you wish, but be sure to answer the following **mandatory** questions:

- What is one thing you learned with respect to each topic covered in this assignment?
- If you were to go back in time and redo this assignment, what would you continue doing? What would you do differently?