CSC421 Assignment 1. Summer 2018 (10pts)

The computer programmer is a creator of universes for which he alone is the lawgiver .... No playwright, no stage director, no emperor, however powerful, has ever exercised such absolute authority to arrange a stage or a field of battle and to command such unswervingly dutiful actors or troops.

Joseph Weizenbaum

`Student Name:` `Student Number:`

`Instructor George Tzanetakis`

Question	Value	Mark
1	1	
2	2	
3	1	
4	2	
5	1	
6	2	
7	1	
Total	10	

1 Overview

The goal of this assignment is to familiarize you with uninformed and informed search algorithms and their implementation. Be aware that some of the questions require much more work than others. Your deliverable will be a report describing your formulation of the problem as a search problem, experimental results and statistics of running various algorithms on random instances of the problem and the code you implemented in order to conduct the experiments. Don't hesitate to contact the instructor via email or utilize the chat room of the ConneX course website for any questions/clarifications you might need.

The questions are marked as (*) necessary to pass, (**) expected and (***) exceptional.

IMPORTANT: YOU SHOULD ONLY SUBMIT A SINGLE PDF FILE WITH YOUR REPORT THROUGH CONNEX. ANY OTHER FORMAT WILL NOT BE ACCEPTED

2 Description

In this assigment the goal is to create multiple search problems corresponding to map search and evalute the performance of different uninformed and informed search algorithms. The search problem simulates a world with many cities. Nearyby cities are joined by roads. Traveling between cities incurs a certain cost. The objective is to find a route between two cities, if it exists, or determine that there is no valid route. The world will be a 100 by 100 2-dimensional grid. In each round of the simulation, 26 cities are uniformly distributed throughout the world (and labeled A through Z). Each city represents a possible state. The starting location is chosen randomly from the list of cities and so is the goal location (excluding the start city).

Each city has a limited number of paths to nearby cities. Paths are bi-directional. Paths are determined by choosing randomly between and 1 and 4 of the closest five cities in terms of Euclidean distance. The path cost between two cities that are joined by a road is equal to the euclidean distance between them.

Your report should contain code snippets showing the different parts that you wrote as well as supporting text and figures describing your answers. I recommend using *Python* but any programming language implementation is fine. You can use the code repository for the AIMA textbook for ideas of how to name your functions and structure your code. You will be expected to understand the code you provide and you might be asked in class to explain it. If you fail to do so I will assume you did not write the code and will change the grade of the corresponding assignment question to 0.

3 Questions

- Formulate this problem as a search problem. Be precise about start state, goal, state-space, succesor-function etc. Describe how you will represent your problem **(*) (1pt)**

- Write a program that generates random instances of the problem. Report on the average number of branches generated. **(*) (2pt)**

- Implement the 3 main uninformed search strategies (Breadth-First Search, Depth-First Search and Iterative Deepening Search) for this problem.

Show examples of calling these functions and the resulting path if found. **(*) (1pt)**

- Report statistics based on solving 100 random instances of the problem using the 3 uninformed search strategies. More specifically for each configuration you will generate 100 random instances (where the start, goal, and roads are randomized as described above) and report the following statistics: average space complexity (maximum number of nodes generated), average time complexity (number of nodes visited), actual running time (in seconds), average path length, and number of problems solved **(**) (2pt)**.
- Conider greedy best-first search as well as A* search with the following three heuristics: (he) straight-line Euclidean distance to the goal state (h0) constant 0 (hm) Manhattan distance to goal state. Explain briefly why these three heuristics are admissable and how they are related to each other in terms of dominance **(*) (1pt)**.
- Implement the two informed search strategies of the previous question and report the same statistics as the uniformed strategies using 100 random instances **(**) (2pt)**.

For an A+ in this assignement undergraduate students can implement any of the following (implementing more than one is ok but will not give you additional points). Graduate students need to implement two out of three to get the A+ point.

- Write code that visualizes the cities and path and show examples of how it would work **(***) (1pt)**.
- Implement bi-directional search using one uninformed and one informed algorithm of your choice. Report the usual statistics and contrast it the standard version. **(***) (1pt)**
- Consider a dynamic scenario where the map changes over time and a fixed random percentage of roads disappears and new roads are added. So the new map is similar to the previous map but different in places. Think about how you could derive an admissable heuristic for the current map that is based on the previous map. This heuristic should would work better than the other three heuristics. Show that this is the case experimentally. **(***) (1pt)**.

4 Deliverable

Your deliverable will be in the form of a report. The report will describe your formulation, implementation and experimental results. For the implementation you can use any programming language you like. The book provides source code in Java, Python, and LISP that you might find useful for this assignment but you are not required to use it.

- **Introduction** In this section you will formulate the problem, and describe in general terms your implementation.
- **Experimental Results** In this section you will present the results of your experiments. The results should be presented in both table and bar graph form. In addition, you should try to comment and interpret your experimental results based on your knowledge of the various search algorithms. For example, do they agree with your expectations.
- **Implementation Overview** In this section you will give an overview of your implementation choices (Programming Languages, structure of your code, conventions, data structures used etc) and any other information you think is relevant for understanding your code.
- **Code listings** You need to provide full source code listings of all the algorithms you implemented. If you based your code on the code provided by the book, you must mention it and still include the relevant code in your report. Your code should be written so that it can be read and understood even by programmers not familiar with the language you are using.

4.1 Grading

The submission should be a single PDF containing code snippets, text and figures with your answers in order. The questions are worth a total of 10 points. Grading will be based on the content of the answers as well as the quality of the report.