CSCE A405 Programming Assignment 1
Due Thursday, September 19, 2019, at 11:59 PM

Purpose: The goal of this assignment is to gain an understanding of the advantages of using informed search techniques.

Requirements: In this assignment, you are required to write a program that solves a variety of 8-Puzzle problems using each of the following search techniques:

a. Breadth-First Search
b. A* Search using the Misplaced Tiles heuristic
c. A* Search using the Manhattan Distance heuristic
d. A* Search using the Gaschnig heuristic

Note that these search techniques differ only in how h(n) is computed. (NOTE: Breadth-First Search uses h(n) = 0.) Your program should allow the user to interactively input the START and GOAL states, and then select one of these four techniques. Your program should then attempt to identify an optimal path from the START state to the GOAL state using the selected search technique. For each search, your program should keep track of the total number of nodes expanded (i.e., "closed"); the maximum number of nodes represented in the search space at any particular time during the search (i.e., the size of the "open list"); and the length of the solution path. Once a solution has been identified, your program should output the each of these values, and display the solution path as a sequence of states from START to GOAL. Your program should allow the user to select a different search technique, and/or enter a different pair of START and GOAL states

The solution space of the 8-Puzzle is symmetrically partitioned according to even or odd PARITY. The parity of a given state can be calculated by counting the number of pairs of tiles in the "wrong" order, assuming the all of the tiles in the following configuration are in the "right" order:

1 2 3
4 5 6
7 8

A solution path exists from a START state to a GOAL state if and only if the same parity (i.e., both even or both odd).

Debug your program using (START, GOAL) state pairs for which a relatively simple solution exists. Once you are satisfied that each search technique works correctly, test your program for several problems having a variety of minimum solution path lengths, and note the relationship between the number of nodes expanded during each search and the length of the solution path. Your test set should also include problems for which the minimum solution path lengths are equal; note the differences in the performance of each search technique on these problems, and identify characteristics that make one problem more difficult to solve than another. Record test results for each search technique.

Work in teams of two or three students. Programs should be written in a high-level language such as Java, Python, or C/C++. Due to the magnitude of this assignment, I strongly recommend that you begin designing, implementing, testing, debugging, and validating the correct operation of your program as soon as possible!

Create a jointly written report summarizing your results. Clearly identify the name and preferred email address of each team member on the cover page of your report. Your report should describe each search technique; provide a summary of test results for each search technique for several representative runs; and identify properties of 8-Puzzle problems that make them relatively harder or easier to solve using each technique. Conclude your report by comparing the overall performance of each of the four techniques.

Submit your report, source code, and executable (if applicable) via Blackboard. I will test your program interactively, read your report, and send your grade with appropriate comments via Blackboard.