CS 461 – ARTIFICIAL INTELLIGENCE

HOMEWORK #3 (5%)

Assigned: Mon Mar 18, 2019

Due: Tue Apr 2, 2019, **2:00 pm NOTE THE UNUSUAL DUE DATE!!!**

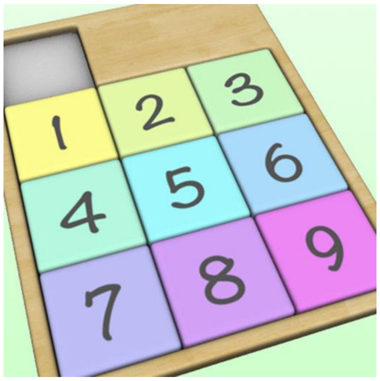
Always indicate the students submitting the homework (at most 5 names). Submit your homework in hardcopy to our TA. (She'll probably give you further instructions.)

Feel free to use any programming language as long as any of you can give a demo using your own computer.

The usual late submission policy applies.

PROBLEM

In this homework, we are again interested in the 9-puzzle. Here's a snapshot of it in its goal state:



Implement the Branch-and-Bound (BBS) and the A* search algorithms (no other algorithms are acceptable) and use them to solve a given instance of the 9-puzzle optimally, that is, with a minimum number of (sliding) moves.

As you know, A* requires admissible heuristics. Use the following function:

- h: Sum of Euclidean distances of the tiles from their goal positions (h is admissible , since in every move, one tile can only move closer to its goal by one step and the Euclidean distance is never greater than the number of steps required to move a tile to its goal position)

Here’s what you must do:

- Run the ‘puzzle generator’ from your preceding homework as many times as necessary to obtain 25 distinct initial states of 9-puzzle.
- Solve each of these 25 instances via BBS and A*.

A submission consists of

- Listing of your program with a clear indication of what parts, if any, of it are borrowed from elsewhere. (I especially encourage you to use an off-the-shelf shortest path algorithm, instead of implementing BBS. In that case, do not list its source code but properly refer to the place where you’ve found it.)
- A graph showing, for your 25 instances (on the x-axis), the number of moves necessary to reach the goal state. Provided that your BBS and A* implementations are correct, the y-axis values must coincide (but either use color or different symbols to distinguish them).

Finally, if you would like to explain your program (you should!), the best place to do so is inside your program (e.g., block comments). Thus, just submit two things: (1) your code listing; (2) a one-page graph, drawn using any package you like.