CMSC 25000 - Homework 2 - Due Wednesday Jan. 28

In this homework you will implement A* search for solving the 8-puzzle. You can either use the state representation and successor functions you defined in the last homework assignment or reimplement them from scratch.

You should turn in:

1. A printout of all your code.
2. A zip file with the code. Use your last name as the name of the file. You should email this file to the instructor in a message with the subject:
   CMSC 25000 homework 2 (your name)
3. A writeup with the results of problem 4 and a brief explanation on how to compile and run the code for problem 5.

Problem 1

Implement a function for computing the Manhatan distance heuristic function for the 8-puzzle. Your function should take a state and return the heuristic value of that state.

Problem 2

Give an implementation of the A* algorithm for solving the 8-puzzle. You should implement the (simpler) tree-search version of the algorithm that does not remember states that have been previously visited.

The main function of your implementation should take a start state and return:
1) The number of nodes generated during the search.
2) The minimum number of moves necessary to solve the puzzle.
3) A shortest path from the start to the goal.

Problem 3

Write a function that takes a number of moves k and generatess a random start state by taking k random actions starting at the goal state. At each step you should pick one of the possible successors of a given state with equal probability. This will generate a start state that is at distance at most k from the goal (but could be less).

Problem 4

Test your A* algorithm by solving random instances of the 8-puzzle generated using the procedure described in Problem 3. For each value of k in $5, 10, 15, 20$ use A* to solve 100 random problems and compute the following as a function of k:

1) The average number of nodes generated in the search

2) The average (shortest path) distance from the start to the goal

Note that (2) will be at most k.

Repeat the experiments with the trivial heuristic function that is constantly zero.

In your writeup you should include 2 tables listing your results with the Manhattan distance heuristic function and with the trivial heuristic.

Problem 5

Write a test program that will let a user type in a start state and then use A* to solve the puzzle from that state. The program should print out the number of nodes expanded during the search, the (shortest path) distance to the goal and the sequence of states in a shortest path to the goal.

The program should prompt the user for a start state and expect a single line of input with a sequence of 9 (unique) digits with no spaces between them:

$$t_0 t_1 \ldots t_8$$

Each value in the input gives the number on a tile (a value of 0 specifies the empty tile) in the following diagram:

$$\begin{array}{ccc} t_0 & t_1 & t_2 \\ t_3 & t_4 & t_5 \\ t_6 & t_7 & t_8 \end{array}$$

For example, the input 475210368 specifies the following starting state:

$$\begin{array}{ccc} 4 & 7 & 5 \\ 2 & 1 & \\ 3 & 6 & 8 \end{array}$$