CSCI 4150 Introduction to Artificial Intelligence, Fall 1999
Assignment 3: out Tuesday September 14, due Thursday September 23

Reading: Chapters 3 and 4
For this assignment, you are to turn in electronically the following procedures: `ep-manhattan`, `ep-heuristic`, (and `ep-fast-heuristic`). Each problem has a written part which is to be handed in on paper in class.

1. (20 points) Russell and Norvig states that breadth first search "is optimal provided the path cost is a nondecreasing function of the depth of the node. (This condition is usually satisfied only when all operators have the same cost.)" (page 74) Using the state space below, explain why this is true.

2. (20 points) Exercise 3.12: Give the time complexity of bidirectional search when the test for connecting the two searches is done by comparing a newly generated state in the forward direction against all the states generated in the backward direction, one at a time.

3. (30 points) We wish to apply the A* search algorithm to find a path from Urziceni to Craiova in the simplified Romanian road map example from Russell and Norvig. Using the road distanced from Figue 4.2 and the straight line distances to Craiova listed below, construct the queue in the A* algorithm:

   (a) after the first node (Urziceni) is expanded
   (b) after the second node is expanded
   (c) after the third node is expanded
   (d) after the fourth node is expanded

   Use the straight line distance as the heuristic. Be sure to show $f(n)$, $g(n)$, and $h(n)$ for each node in the queue, and indicate which node has been expanded at each step.

Arad	290	Mehadia	110
Bucharest	170	Neamt	330
Craiova	0	Oradea	350
Dobreta	120	Pitesti	138
Eforie	350	Rimmnicu Vilcea	146
Fagaras	190	Sibiu	200
Giurgiu	140	Timisoara	225
Hirsova	325	Urziceni	240
Isai	350	Vaslui	350
Lugoj	140	Zerind	310

4. On the course home page, there will be a fairly efficient implementation of the A* algorithm. You will call the search procedure with four arguments:

```
(astar start-state goal-state get-children heuristic)
```

The start and the goal state are lists of 9 elements which represent the state of the eight-puzzle as described in the previous assignment. You will use the `ep-children` function you wrote for the previous assignment as the `get-children` argument. You must also provide a heuristic function which takes two arguments: the current state and the goal state (in that order). In this problem, you will write and experiment with several different heuristic functions.

(a) (10 points) Write a function `(ep-manhattan current goal)` which the sum of the Manhattan distance for each of the eight tiles from the current state to the goal state. For example:

```
(define ss1 '(space 1 2 3 4 5 6 7 8))
(define ss2 '(1 2 3 4 5 6 7 8 space))
(ep-manhattan ss1 ss2) ==> 12
```

You may want to use the `ep-distance` function you wrote for assignment 2.

(b) (40 points) Experiment with the `ep-manhattan` heuristic and then write your own admissible heuristic `(ep-heuristic current goal)`. You should try to attain better performance (i.e. fewer nodes searched) on average than the manhattan distance sum. Note that your heuristic may not outperform the manhattan distance sum heuristic on every test case; you should try several different start and goal states.

Explain why your heuristic is admissible. Is your heuristic monotonic? Why or why not?

(c) (5 points) Attempt this part only if you have finished the rest of the assignment!

Write a heuristic `(ep-fast-heuristic current goal)` that is designed to guide the A* search to a solution, not necessarily optimal, as quickly as possible. This heuristic need not be admissible. Explain your heuristic.