Homework #1

Due Date: September 25th, 9PM

Submit the zip file on Blackboard by the deadline

1. **Programming Assignment**: The goal of this assignment is to develop a program, in either JAVA, C++, or PYTHON, that implements search algorithms for solving a nineteen-tile puzzle. The search-based solution must follow the strategy of using a priority queue for the **Open List** and another data structure of your choice for the **Closed List**. The main steps of the search algorithm must be the same as described in class, that is, a repetition of the following steps until the Goal State is discovered:

    a. Remove the highest priority state *S* from the Open List

    b. Check to see if this state *S* is the goal state. If *S* is the goal state then output the path from the start state to the goal state, print the relevant information of the solution as outlined below, and stop the search algorithm.

    c. (If *S* is not the goal state, then) Find all successor states of the state *S*, compute their priority values, and insert them into the Open List. Place the state *S* on the Closed List. If a newly generated state already exists on the Closed list then do not add this to the Open list. If a newly generated state already exists. On the Open list, then set the path cost for that state to the lower of the costs given in the two states (and adjust its priority in the open list appropriately).

    Each state should be a structure (object?) containing the following data fields:

    - State ID (An integer Number)
    - State ID of the Parent State (From which this state was generated)
    - A 1X20 vector describing the current 19-tile puzzle board. For example, the board shown here translates to the vector representation shown below:

5	19	7	8
9	11	10	3
14	6	18	16
12		2	13
4	15	1	17

[5 19 7 8 9 11 10 3 14 6 18 16 12 **0** 2 13 4 15 1 17]

Notice that the blank slot on the board is represented by a ‘0’ in the State-vector.

- The g(n) value for the state.

- The h(n) value for the state
- The f(n) (= g(n)+h(n)) value of the state.
- Priority value assigned to this state by your algorithm.

The characteristics of your program must include the following:

a. The program takes as inputs two vectors, the first one describing a start state and the second one describing a goal state. For example, the input to the program may look like:
   [5 15 7 8 9 11 10 3 12 0 2 13 4 14 1 6 16 17 18 19] [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 17 16 19 18]
b. The output of the program must include the following items:
   i. A sequence of State descriptions in which the first state is the start (or the goal) state and the last state is the goal (or the start) state as obtained by the Breadth First Search Algorithm. Show the complete state description including the ID's, g(n), h(n) and f(n) values.
   ii. Length of the path between the start and the goal state in terms of the number of moves; the count of nodes added to the Open List during the search process; and the count of nodes added to the Closed List during the search process (BFS Search).
   iii. A sequence of State descriptions in which the first state is the start (or the goal) state and the last state is the goal (or the start) state as obtained by the A* algorithm using the h1 heuristic (described below). Show the complete state description including the ID's, g(n), h(n) and f(n) values.
   iv. Length of the path between the start and the goal state in terms of the number of moves; the count of nodes added to the Open List during the search process; and the count of nodes added to the Closed List during the search process (A* Search using heuristic h1).
   v. A sequence of State descriptions in which the first state is the start (or the goal) state and the last state is the goal (or the start) state as obtained by the A* algorithm using the h2 heuristic (described below). Show the complete state description including the ID's, g(n), h(n) and f(n) values.
   vi. Length of the path between the start and the goal state in terms of the number of moves; the count of nodes added to the Open List during the search process; and the count of nodes added to the Closed List during the search process (A* Search using heuristic h2).

c. The cost of making a move for the game is as follows: Whenever a tile numbered 1 through 9 is moved the cost of making the move is 1 unit. Whenever a tile numbered 10 through 19 is moved the cost of making the move is 2 units.

d. The heuristic function h1 for the A* search is as follows: The estimated cost of taking a board B to the goal state G is the number of tiles in B that are not in the correct location as required by G.

e. The heuristic function h2 for the A* search is as follows: The estimated cost of taking a board B to the goal state G is the **sum** of the smallest number of moves for each tile, that is not at its final location, to reach its final location as required by G.

The following items, bundled in a single zip file, must be submitted on Blackboard in response to this assignment:

a. Source code file for your program, named <Your First Initial><Your Middle Initial><Your Last Initial>'SourceCodeFile'.___ Your program must be written in one of these three languages: JAVA, PYTHON, or C++.

b. A README.pdf file listing the command line instructions needed for (i) compiling your program, and (ii) executing your program with two 1X20 vectors as inputs.

c. Output of your program obtained for the following three cases included in a file named: <FI><MI><LI>.OutPutFile.pdf:

   i. [16 17 5 1 3 4 2 10 6 8 13 9 7 12 0 14 11 15 18 19] [16 17 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 18 19]

   ii. [1 0 3 4 5 2 7 8 9 6 15 11 13 10 14 12 16 17 18 19] [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 16 17 18 19]

   iii. [2 0 3 4 1 5 7 8 9 6 10 12 13 14 11 15 16 17 18 19] [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 17 16 18 19]

Grading of the project will be be out of 100 points, distributed as follows:

a. 40 points for the reasonableness of your outputs submitted along with the submission. (There is no one single correct output).

b. 10 points for the readability of the output file. That is, there should be a text description of what each number is and what input corresponds to which output, etc.

c. 10 points for the comments in the source code that describe the role of each of the sections of the code and their role/contribution in the overall program.

d. 40 points for the reasonableness of the outputs obtained for three different test inputs for which your algorithm will be executed and tested by the TA.

Do not throw away this program; an additional homework assignment will be given to design a new heuristic for a slightly modified problem situation.