Homework 3

Due: Sep 22, 2014 - 2pm

Instructions for submitting programs

The beginning of the program should contain comments with your name. You should submit .zip folder* on blackboard with following items.

1. All source code.
2. A readme file: explaining how to compile and run your program.
3. A PDF file including 3 test cases(inputs used and comparisons with last assignemnt)

   * - Name of folder should be your netID

Problem: In this assignment you will implement a program that uses A*(using two heuristics given in book) search to solve the Fifteen Puzzle game. Your program will take an initial board position, and then try to find a sequence of moves from the initial position to the goal position. Moves are represented by moving the blank space left, right, up, or down. For example, suppose we wanted to solve the puzzle shown below. This could be accomplished by the following sequence of moves (we represent a move as the direction the empty space moved):

1 2 3 4		1 2 3 4		1 2 3 4
5 6 7 8		5 6 7 8		5 6 7 8
9 10 0 11	**R**	9 10 11 0	**D**	9 10 11 12
13 14 15 12		13 14 15 12		13 14 15 0
initial				goal

where R=right, L=left, U=up, D=down. So we can solve this puzzle with the sequence of moves "RD".

Keep track of the number of expanded nodes and computation time. Compare the running time and memory usage of A* to uninformed search done in last assignment.

Run your program and fill in the following table. Mention in your PDF file which initial state you choose(3 initial board test cases should be included).

Test case 1:

Search Strategy	**Memory usage**	**Running time**	**Number of expanded nodes.**	**Path to goal node.**
BFS				
ID-DFS				
A* Heuristic 1				
A* Heuristic 2				

Test case 2: Table

Test Case 3: Table

Game Board:

For example, the above puzzle board is given by user(command prompt) as "1 2 3 4 5 6 7 8 9 10 0 1113 14 15 12"

Your implementation: The code for this assignment must be written in C , C++ or Java. It must be well documented.

Output:

Your program should output moves used to reach goal state(by both heuristics:)

Heuristic 1: Output

Heuristic 2: Output

Output should be of following format

board	**|**	**number of moves |**	**solution**
1 2 3 4 5 6 7 8 9 10 0 11 13 14 15 12		2	R D

Online Code Repository

You can use AIMA Code if you want.

http://aima.cs.berkeley.edu/code.html

https://code.google.com/p/aima-java/downloads/list