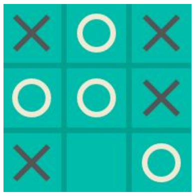
COSC 4368: Artificial Intelligence Spring 2018
Homework # 1:

Assigned Jan 31st, 2018, **Due** Feb 7th, 2018 (midnight)

Overview

This assignment consists of some short answer questions and a programming assignment. You need to upload a zipped file consisting of a pdf and source code file. Your report should cover the answers to questions 1,2,3 and a short report of how you implemented the program. Submit your zipped file on Blackboard.

General Questions



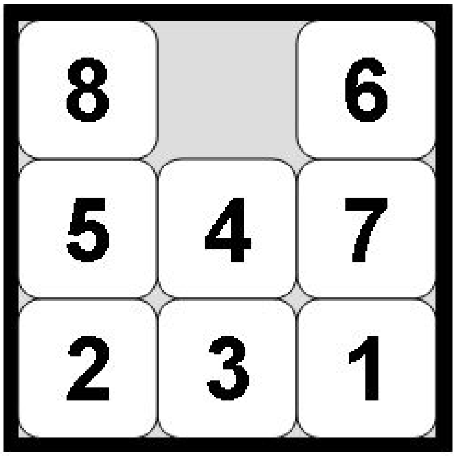
X	O	X
O	O	X
X		O

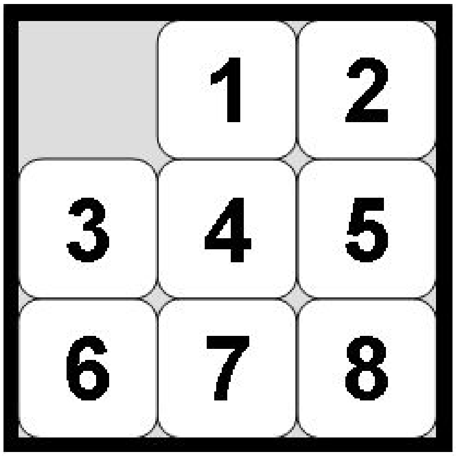
1. Consider the state of a tic tac toe board in the image above, assuming X begins first, count how many possible ways are there to begin a game and reach this state. ***(10 pts)***

2. Consider a program which plays tic tac toe and uses one of the search algorithms you have seen in class to search for terminal states (i.e., win, draw, loss). The root node (no marks on board) is at a depth level of 0. ***(10 pts)***
    a. What is the maximum depth level of a terminal state?
    b. What is the minimum depth level of a terminal state?
    c. What is the minimum depth level of a terminal loss?
    d. What is the minimum depth level of a terminal win?

3. Suppose you are implementing a tic tac toe AI with a heuristically-guided technique. The board is represented by an array of 9 cells where -1 represents X, 1 represents O, and 0 represents a free space, design a heuristic function H(B, P) that takes as parameters, the board B, and the Player P(-1 or 1), and returns how well that player is doing. Explain briefly the intuition for your function and discuss its rationale. **(10 pts)**

Programming Assignment





Objective (70 pts)
Write a program to solve the 8-puzzle problem using the A* search algorithm (you can use Java, Python or C/C#/C++ for implementation).

The program should first randomly shuffle the board with a sequence of 5 to 20 random moves, then pass the board to the AI module which would then return a sequence of moves leading to the goal state. If a solution is found, your program will then perform the solution of moves to obtain the goal state.

Output
Sample output on next page. You should print the board each time a move is performed during the shuffling, and when you perform the solution. The whole set of moves done when running the algorithm (A*) do not need to be printed out; just print the final sequence of steps to reach the goal. The board should be printed as a grid of 3 by 3 numbers.

```
Shuffling board:
+board+
|0|1|2|
|3|4|5|
|6|7|8|
+-----+
+board+
|1|0|2|
|3|4|5|
|6|7|8|
+-----+
+board+
|1|2|0|
|3|4|5|
|6|7|8|
+-----+
+board+
|1|0|2|
|3|4|5|
|6|7|8|
+-----+
+board+
|1|2|0|
|3|4|5|
|6|7|8|
+-----+
+board+
|1|2|5|
|3|4|0|
|6|7|8|
+-----+
+board+
|1|2|5|
|3|4|8|
|6|7|0|
+-----+
+board+
|1|2|5|
|3|4|8|
|6|0|7|
+-----+
+board+
|1|2|5|
|3|4|8|
|6|7|0|
+-----+
shuffle done, attempting to solve
+board+
|1|2|5|
|3|4|8|
|6|7|0|
+-----+
+board+
|1|2|5|
|3|4|0|
|6|7|8|
+-----+
+board+
|1|2|0|
|3|4|5|
|6|7|8|
+-----+
+board+
|1|0|2|
|3|4|5|
|6|7|8|
+-----+
+board+
|0|1|2|
|3|4|5|
|6|7|8|
+-----+
Solution length:  4 moves
Shuffle length:  8 moves
```