**California State University, Fresno – Fall 2013**

**Computer Science 166, Principles of AI (3 units)**

**Assignment #3**

**Due:** 1/30/14

**Value:** 30 points

**Part 1: State-Space Search: 8-Puzzle**

In this assignment you will explore state-space search with the 8-puzzle domain. You will be using the python code in the file puzzle8.2.py. You will explore two different search algorithms to the problem. Breadth-First Search and Iterative-Deepening Depth-First Search. To change the search algorithm used you will need to modify the code in main for the variable ‘searchFn’:

# Search Function Choice

#searchFn = breadth\_first\_tree\_search

searchFn = iterative\_deepening\_search

.

Comment out the search algorithm not being used, and uncomment the one being used. You can modify the start state by setting the variable ‘startstate’ to the initial state. Several example start states are already defined. See example code below:

# Start States

s1 = [[1,2,3],[4,5,6],[0,7,8]]

s2 = [[1, 3, 0], [4, 2, 5], [7, 8, 6]]

s3 = [[0, 1, 5], [7, 3, 6], [2, 4, 8]]

s4 = [[7, 1, 5], [0, 3, 6], [2, 4, 8]]

s5 = [[7, 1, 5], [2, 3, 6], [0, 4, 8]]

examplePuzzle = [[2,1,6],[0,4,8],[7,5,3]]

startstate = s1

Run both search algorithms on each of the start states and document the length of the solution (if one is found) and the number of search nodes expanded.

**Part 2: Iterative-Deepening Search**

Find and list the code for Iterative-Deepening Search within the file ‘search.py’ and briefly explain the search algorithm.

**Part 3: Human Solution**

Solve the problem from the start state s5 and ‘examplePuzzle’ and document the length the number of moves you required. Did you find the optimal solution?

**Part 4: Heuristic Search**

Describe heuristic search and explain how it can be used to improve the performance of Iterative-Deepening Depth-First search through Iterative-Deepening Astar search.