CSC 330 – Artificial Intelligence
Program Assignment #2 – Uninformed and Informed Searching
Due Wednesday, February 21, at 5:00 PM

Description:

This assignment has you writing two search algorithms to solve the eight-puzzle problem. The program should determine the path from a given starting state to the goal, as well as how many nodes were expanded during the search. This will give you some experience with coding search algorithms, as well as (hopefully) allowing you to see the benefit of informed search algorithms.

Eight-Puzzle:

The eight-puzzle is a physical puzzle that involves sliding tiles on a 3x3 grid containing 8 tiles and an empty spot. You need to slide tiles around in the puzzle until you reach the solution:

1	2	3
8	0	4
7	6	5

The empty spot will be represented by a 0. When determining what moves are possible, you can only slide tiles adjacent to the empty spot into the empty spot. It is usually easier to consider the move as moving the empty space, rather than the tile.

Expectations:

Your program should implement two search algorithms to solve the eight-puzzle. One of those algorithms must be an uninformed search algorithm (BFS, DFS, or IDS); the other must be an informed search algorithm (best-first or A*).

When completed, your program should do the following:

- Print a welcome message and inform the user what the program is doing.
- Prompt the user to enter a file name containing an eight-puzzle that you wish to solve. This file will contain a 3x3 grid of digits. If the file does not load properly, print an error message and end the program. Some example files are provided on the Moodle site.
- Print the initial state of the puzzle.
- Prompt the user to select which algorithm they wish to run on this puzzle. If you are implementing DFS, also prompt the user for a depth limit.
- Run the selected algorithm on the initial state.
- When the algorithm is finished, the program should print the following information:
    - Print all the states in the path from the start state to the goal state.
    - Print the length of that path (the number of moves required to solve the puzzle).
    - Print the total number of nodes expanded while doing the search.

Specifications:

Your program must meet the following specifications:

1. Your program represents a sincere attempt to solve the problem.
2. Your program compiles and runs, assuming specification 1 is met.
3. Your program successfully loads and prints information from the file entered by the user.
4. Your program allows the user to select from two different algorithms.
5. Both your algorithms successfully solve the very_easy.txt example file.
6. Both your algorithms successfully solve the easy.txt example file.
7. Both your algorithms successfully solve the medium.txt example file.
8. Both your algorithms successfully solve the hard.txt example file.
9. Both your algorithms successfully solve the very_hard.txt example file.
10. Both your algorithms successfully solve a test file of my own devising.
11. Both your algorithms successfully solve a second test file of my own devising.
12. Your program correctly prints the path to the goal for all tests.
13. Your program prints the correct length of the path to the goal for all tests.
14. Your program prints the correct number of nodes expanded for all tests.
15. Your program is well-written and styled (formatting, consistent indentation, comments, etc.).