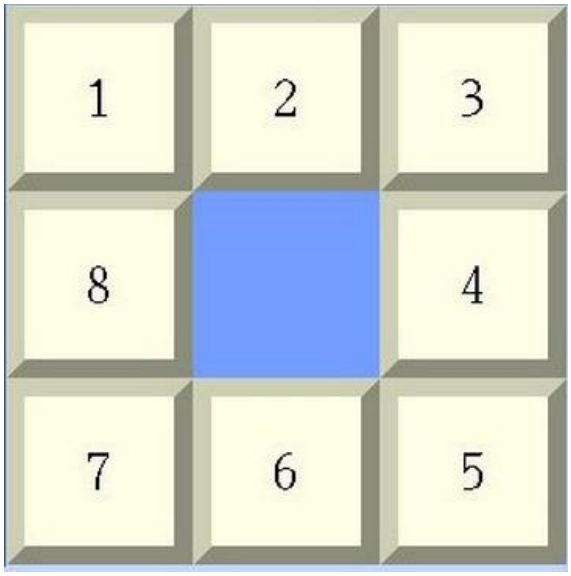
CSCI561 – Introduction to Artificial Intelligence
Instructor: Dr. K. Narayanaswamy

Assignment 1 – 8 Puzzle
Due: 9/19/2011 11:59:59pm
Electronic Submission (on Blackboard)

1. Problem Definition

The 8-puzzle is a sliding puzzle that consists of a grid of numbered squares with one square missing, and the labels on the squares jumbled up. The goal of the puzzle is to un-jumble the squares by only making moves which slide squares into the empty space, in turn revealing another empty space in the position of the moved piece.



Input:

2 8 3
1 6 4
7 0 5

Note - 0 means the blank square.
The output file would be like this:

1 2 3
8 0 4
7 6 5

And Output screen should display the following information:

Sequence of tiles to be moved:

1st Move: Blank Tile Up
2nd Move: Blank Tile Up
3rd Move: Blank Tile Left
4th Move: Blank Tile Down
5th Move: Blank Tile Up

Number of moves: 5

2. Algorithm Requirement

You will implement A* algorithm for this assignment. A* requires admissible heuristics in order to succeed. For a discussion of admissible heuristics for this kind of tiling problem see Lecture Notes or the Text Book.

3. Coding Requirement

Your program must be written in C++ or Java. You may use the Standard Libraries (e.g. STL for C++ or Java's standard container libraries), you may not use any 3rd party objects/algorithms without prior permission from the course TA or Professor (don't go download code that runs search algorithms). No collaboration will be allowed on this project (either the program or the questions). These must reflect just your own work, with no outside help (except for questions asked of the instructor and TA). This explicitly includes no use of any code from elsewhere (including the Internet) without explicit permission from the instructor or TA.Your program must compile and run on Aludra (see: http://www.usc.edu/its/web/getting started/ppages.html if you are unfamiliar with Aludra).

4. Project Questions

Question: 1[5 points]:
Describe the heuristic you used, and justify the fact that it is admissible.

Question: 2[10 points]:
While we guarantee that the inputs we provide to test your program will indeed lead to solutions, it is well known that not all inputs are solvable for the 8-puzzle problem. Research this aspect of the 8-puzzle problem and provide an answer to the following question: How would you modify your program so that it automatically detects those cases of inputs which would never yield a solution? The answer must be sufficiently detailed (no more than 1 page) to illustrate how your detection process would work.

Question: 3[5 points]
How did you test your code? Do not simply enumerate test cases, but describe how you went about testing for specific algorithmic issues.

5. Submission Guidelines

• Code

For the C++ programmers:

Make sure that your source code can be compiled and executed on aludra.usc.edu. Please provide a README file with instructions to run your programs and a Makefile to compile where possible.

For the JAVA programmers:

All JAVA classes should be in package CS561A1.<lastname> and the source files should be in src/CS561A1/<lastname>

Place all the required files in a single directoryZip all the files in your submission into a single file called:

Firstname_LastName_A1.zip

Only Zip format will be accepted. Please do not use any other processes or tools.

• Execution Details

puzzle –i inputFile –o outputFile

or

java –cp classes CS561.<lastname>.puzzle –i inputFile –o outputFile

inputFile indicates the name of the input File, and outFile is the text file which will enclose the solution

• CS561A1Written.pdf

A pdf containing your answers to the written questions. Please number your answers exactly as we number our questions.

• README.txt

This file must contain the following information
– Your name (First and Last)
– Your USC Student number
– Your USC email address
– Implementation Details
– Any additional information you want us to know when grading your work.

6. Grading Guidelines [2 + 60 + 18 + 5 + 10 + 15 = 100]

• File names and extensions (case sensitive) are all correct - 2 pts

• Your program will be tested with 3 different input files. Grades will be assigned based on the following:

1. Correct listing of optimum movement sequence for each input: 20 * 3 - [60 points]
2. Correct number of steps for each input: 6 * 3 - [18 points]

• Project Questions: Question 1 - [5 points], Question 2 - [10 points], Question 3 - [5 points]

• Missing README file: [-10 points] (even if lots of comments in code)

• Your code must be robust. If your program crashes or fails to terminate in reasonable time on any test case, then it fails that test. If your program terminates gracefully (for example saying "error on line 2 of input file"), more partial credit might be awarded.

Please make sure ALL source files are submitted in the zip file you submit. Errors in submission will be assessed –25 points.

A program that does not compile as submitted will be given 0 points. None of your other submissions will be graded if the program does not compile.

Only your FINAL submission will be graded.

For policies on late submissions, please see the Syllabus from the course home page. These policies will be enforced with no exceptions.