

# CS 380: Assignment 1

## Fall 2006

### Written

Due before class on 2006-10-06.

From R&N Chapters 2 & 3: Exercises 2.2, 2.6, 3.7

What is the average branching factor of 8-puzzle<sup>1</sup>?

*Extra Credit:* What is the average number of “slides” required to solve an 8-puzzle?

Please show all work.

### Programming

Due before class on 2006-10-13.

You will be implementing a solver for the 8-puzzle game. You will use uninformed search and must implement BFS, DFS, and DLS.

Once you have implemented all three search techniques, create a short report (just a paragraph or two) comparing their efficiency. To be statistically sound, this will require running each algorithm a number of times over randomly-generated 8-puzzle instances. Either as a part of this report or a separate README should be instructions for running/testing your code.

The game logic has been implemented in `8puzzle.lisp` (available on the course website); you may choose to use this file as a starting point.

All work must be done in Lisp and be executable in `clisp`.

You should ultimately end up with a function called something like “`solve-8puzzle`” that, given an initial 8-puzzle instance will return a list of the moves required to get to the goal state. For example:

```
[1]> (load "8puzzle.lisp")
;; Loading file 8puzzle.lisp ...
;; Loaded file 8puzzle.lisp
T
[2]> (let ((puzzle (random-puzzle))) (print-puzzle puzzle) (solve-8puzzle 'BFS puzzle))
```

```
-----
| 1 | 6 | 2 |
-----
| 4 |   | 3 |
-----
| 7 | 5 | 8 |
-----
```

(UP RIGHT DOWN LEFT DOWN RIGHT)

Here are some hints/guidelines:

- Implement the GENERAL-SEARCH algorithm described in R&N (this means you’ll need the functions MAKE-QUEUE, EMPTY?, *et cetera*).

---

<sup>1</sup>For a description of 8-puzzle see <http://en.wikipedia.org/wiki/N-puzzle>

- Implement the queuing functions for BFS, DFS, and DLS.
- You will probably need a function to determine whether the goal state has been reached and a function that generates successor states.
- The search-space of 8-puzzle is cyclic; you will probably need some way of making sure equivalent states are not expanded multiple times.
- Be aware that these algorithms take a long time to run. Depending on the speed of your processor and the randomly-generated 8-puzzle instance, DFS may take an inordinate amount of time to terminate. You may wish to read about configuring the Lisp compiler (compiled Lisp code will run orders of magnitude faster than interpreted code).