

# Final Exam for CS392 X1

Hongwei Xi

May 5th, 2025

As always, if you need clarification, please do not hesitate to ask for it!

## Problem 1 (100 points)

A solution to the 8-queen puzzle can be found by visiting the following link:

<https://ats-lang.sourceforge.net/DOCUMENT/INT2PROGINATS/HTML/x631.html>

Please *translate* this solution into Java. Note that this solution is functional in the sense that it makes no use of updates (to variables). Your translation in Java should make no use of variable updates as well (except in constructors). Also, you should make no use of loops (either while-loops or for-loops); instead, you can use recursive methods in place of loops.

## Problem 2 (100 points)

During lectures, we talked about implementations of DFS and BFS, where the former uses a stack (FILO) for storage while the latter uses a queue (FIFO). Please implement two generic classes DFSforCS392 and BFSforCS329 which implement DFS and BFS, respectively. You have the freedom to decide what methods you want in these classes.

## Problem 3 (100 points)

Please give a high-level description in English as to how Problem 1 (8-queen puzzle) can be solved using either DFS or BFS.

- Please give a DFS-based implementation according to your description that should directly use the DFSforCS392 class (see Problem 2).
- Please give a BFS-based implementation according to your description that should directly use the BFSforCS392 class (see Problem 2).

## Problem 4 (100 points)

A description on Game-of-24 and an accompanying demo can be found by visiting the following link:

<https://github.com/xanadu-lang/xats2js/tree/master/docgen/P-Ground/githwxi/Game-of-24>

Please give a high-level description in English as to how Game-of-24 can be solved using either DFS or BFS.

- Please give a DFS-based implementation according to your description that should directly use the DFSforCS392 class (see Problem 2).
- Please give a BFS-based implementation according to your description that should directly use the BFSforCS392 class (see Problem 2).

## Problem 5 (100 Bonus points)

A description on *Knight's tour* can be found on the following wiki page:

[https://en.wikipedia.org/wiki/Knight%27s\\_tour](https://en.wikipedia.org/wiki/Knight%27s_tour)

For a chess board (of dimension  $8 \times 8$ ), you may need Warnsdorf's rule (a heuristic) to find a Knight's tour from a given board position. One way to implement this heuristic is by doing PBS (priority-based search) (instead of plain DFS or BFS), where you use a priority queue (e.g., heap-based) to store intermediate nodes.

(If you can solve this problem, then I will be happy to write a glowing reference letter for you in case you need it (e.g., for graduate school application))