

Introduction to Program Synthesis - Exercise III

Search Methodologies

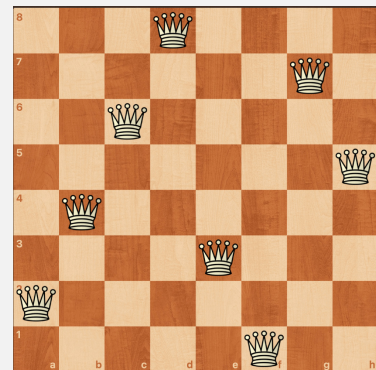
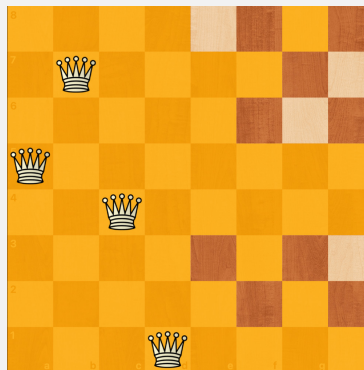
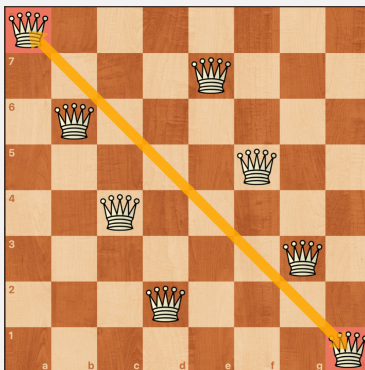
Deadline: 30 Jan 2025 (12 pm)

In these exercises, you will apply efficient search algorithms to well-known combinatorial problems.

Exercise 1: Chess - Queen's Puzzle

The Queen's Puzzle in chess requires you to place eight Queens on an 8x8 chess board, such that no two queens are able to "see" each other. Queens can look infinitely far in horizontal, vertical, and diagonal direction. See the diagram below for examples.

- (1) How large is the naive search space for the Queen's Puzzle on an 8x8 chess board?
- (2) Implement the Queen's Puzzle in Python. Build a simple random search that tries random solutions until a feasible one is found.
- (3) We know that random search is far from an ideal search algorithm for this problem. Use the Backtracking Enumerative Search algorithm you learned in the lecture, to find **all** feasible solutions to the Queen's Puzzle. (*Hint: There are in total 92 distinct solutions.*)



Left: Infeasible solution. The top left and bottom right queens are seeing each other.
Middle: A partial solution, showing which squares are still available for the remaining queens.
Right: A correct solution.