1 n-Queens

Max Friedrich William Bezzel invented the “eight queens puzzle” in 1848: place 8 queens on a chess board such that none of them can capture any other. The problem, and the generalized version with \( n \) queens, has been studied extensively (a Google Scholar search turns up over 3500 papers on the subject).

Queens can move any number of squares along rows, columns, and diagonals (left); An example solution to the 4-queens problem (right).

(1) Formulate \( n \)-queens as a search problem:

Start State:

Successor Function:

Goal Test:

(2) Like many search problems, \( n \)-queens can be parameterized in a number of different ways. Suggest a different formulation (hint: how many successors does your function return?)

(3) How large is the state space in your formulations?

(4) Why is depth first search preferable to breadth first search for this problem?
2 8-puzzle

The 8-puzzle (actually, its more complex cousin, the 15-puzzle) was briefly the subject of a gaming craze in the U.S. in 1880 after Sam Lloyd offered $1000 for a solution to what turned out to be an impossible tile arrangement. The puzzle involves sliding tiles until they are ordered correctly. To solve these puzzles efficiently with A* search, good heuristics are important.

\[
\begin{array}{|c|c|c|}
\hline
1 & 2 & 3 \\
\hline
5 & 6 & 7 \\
\hline
9 & 10 & 11 \\
\hline
13 & 15 & 14 \\
\hline
\end{array}
\]

Sam Lloyd’s unsolvable 15-puzzle. Why can’t it be solved?

(1) Create a heuristic for the 8-puzzle based on the number of misplaced tiles.

(2) Create a heuristic using Manhattan Distance.

(3) Explain why your heuristics are admissible.