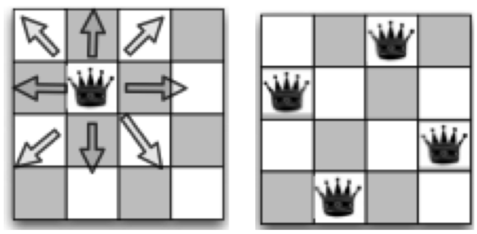
CS 188 Fall 2017 Section 0 & 1: Search

1 n-Queens

Max Friedrich William Bezzel invented the eight queens puzzle in 1848: place 8 queens on an 8×8 chess board such that none of them can capture any other. The problem, and the generalized version with n queens on an $n \times n$ chess board, 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).

a) Formulate n-queens as a search problem. Have each search state be a board, where each square on the board may or may not contain a queen. To get started, we'll allow boards in our state-space to have any configuration of queens (including boards with more or less than n queens, or queens that are able to capture each other).

Start State: An empty board

Goal Test: Returns True iff n queens are on the board such that no two can attack each other

Successor Function: Return all boards with one more queen placed anywhere

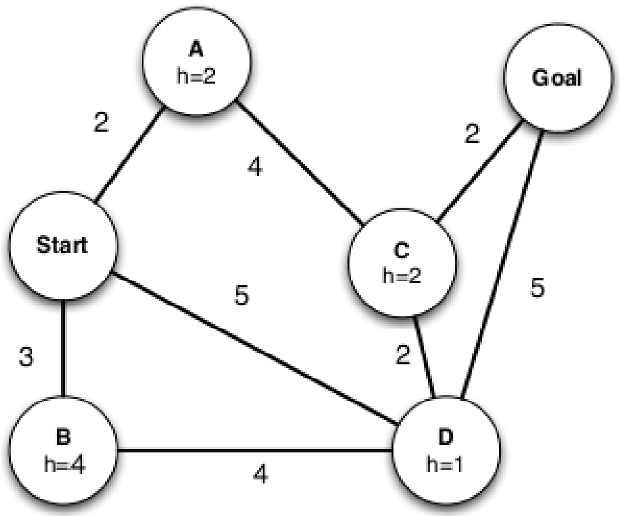
b) How large is the state-space in this formulation? 2^{n^2}, or 1.8×10^{19} for 8-queens.

c) One way to limit the size of your state space is to limit what your successor function returns. Reformulate your successor function to reduce the effective state-space size. The successor function is limited to return legal boards. Then, the goal test need only check if the board has n queens.

d) Give a more efficient state space representation. How many states are in this new state space? A more effective representation is to have a fixed ordering of queens, such that the queen in the first column is placed first, the queen in the second column is placed second, etc. The representation could be a n-length vector, in which each entry takes a value from 1 to n, or "null".

Since each of the n entries in the vector can take on $n+1$ values, the state space size is $(n+1)^n = 4.3 \times 10^7$ for $n = 8$.

2 Search algorithms in action



For each of the following graph search strategies, work out the order in which states are expanded, as well as the path returned by graph search. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. The start and goal state are S and G, respectively. Remember that in graph search, a state is expanded only once.

a) Depth-first search.
States Expanded: Start, A, C, D, B, Goal
Path Returned: Start-A-C-D-Goal

b) Breadth-first search.
States Expanded: Start, A, B, D, C, Goal
Path Returned: Start-D-Goal

c) Uniform cost search.
States Expanded: Start, A, B, D, C, Goal
Path Returned: Start-A-C-Goal

d) Greedy search with the heuristic h shown on the graph.
States Expanded: Start, D, Goal
Path Returned: Start-D-Goal

e) $A^\star$ search with the same heuristic.
States Expanded: Start, A, D, B, C, Goal
Path Returned: Start-A-C-Goal