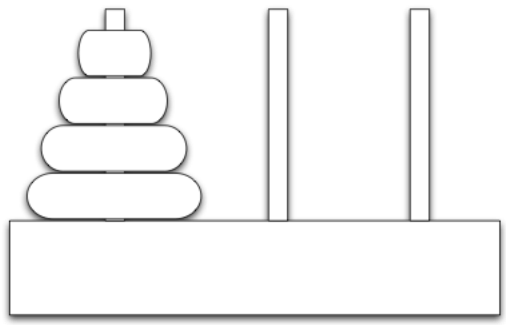
CS 188 Summer 2019 Section 1: Search

1 Towers of Hanoi



The Towers of Hanoi is a famous problem for studying recursion in computer science and recurrence equations in discrete mathematics. We start with N discs of varying sizes on a peg (stacked in order according to size), and two empty pegs. We are allowed to move a disc from one peg to another, but we are never allowed to move a larger disc on top of a smaller disc. The goal is to move all the discs to the rightmost peg (see figure).

In this problem, we will formulate the Towers of Hanoi as a search problem.

(a) Propose a state representation for the problem

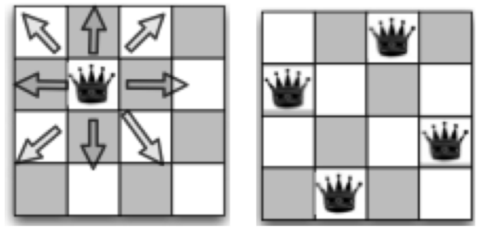
(b) What is the start state?

(c) From a given state, what actions are legal?

(d) What is the goal test?

2 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:

Goal Test:

Successor Function:

b) How large is the state-space in this formulation?

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.

d) Give a more efficient state space representation. How many states are in this new state space?