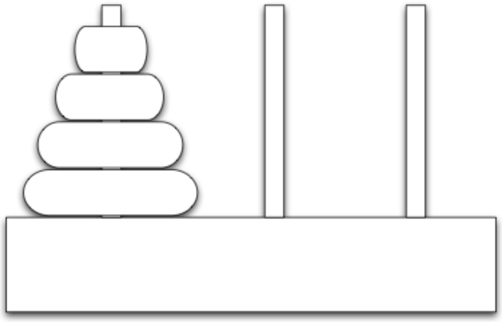
CS 188 Fall 2018 Section 1: Search

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

One possible state representation would be to store three lists, corresponding to which discs are on which peg. If we assume that the N discs are numbered in order of increasing size $1, ..., n$, then we can represent each peg as an ordered list of integers corresponding to which discs are on that peg.

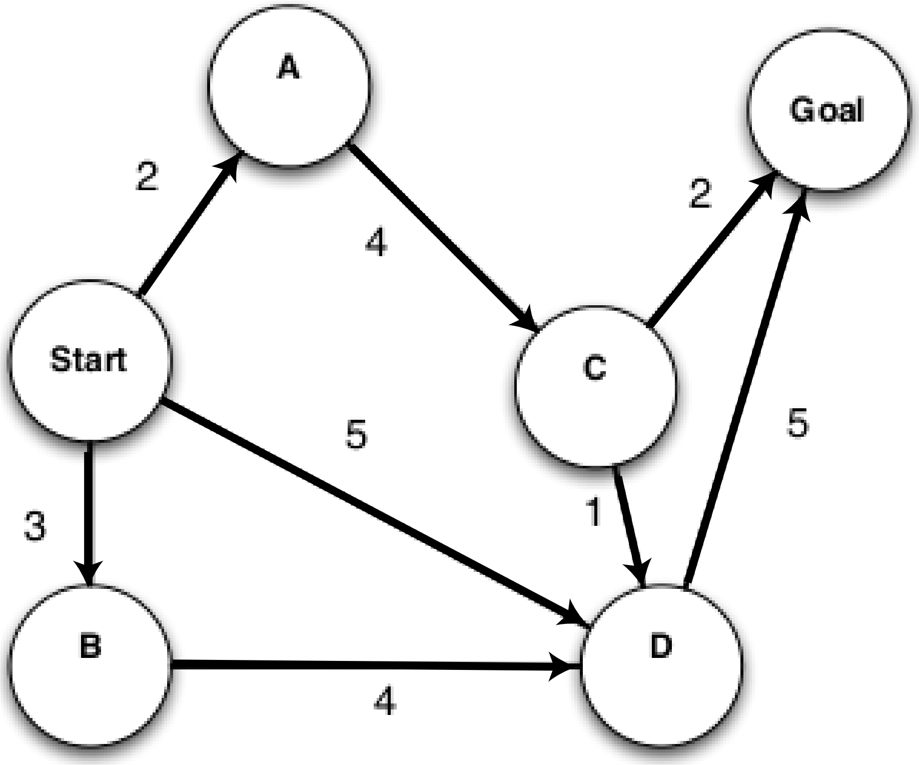
(b) What is the start state? $([1, ..., n], [], [])$

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

We can pop the first integer from any list (i.e., peg) and push it onto the front of another list (peg), so long as it is smaller than the integer currently at the front of the list being pushed to (i.e., peg being moved to).

(d) What is the goal test? Is the state the same as $([], [], [1, ..., n])$?

2 Search algorithms in action



For each of the following search strategies, work out the path returned by the search on the graph shown above. 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.

a) Depth-first search.
Path Returned: Start-A-C-D-Goal

b) Breadth-first search.
Path Returned: Start-D-Goal

c) Uniform cost search.
Path Returned: Start-A-C-Goal