Exercises Lecture 3
Intelligent Systems Programming (ISP)

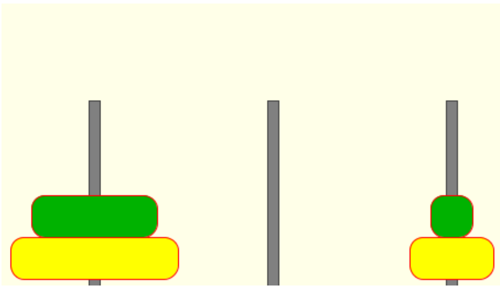
Exercise 1

The Towers of Hanoi is a mathematical game or puzzle. It consists of three pegs, and *n* disks of different sizes which can slide onto any peg. The puzzle starts with the disks neatly stacked in order of size on the left-most peg, smallest at the top, thus making a conical shape.

The objective of the game is to move the entire stack to the right-most peg, obeying the following rules:

1. Only one disk may be moved at a time.
2. Each move consists of taking the upper disk from one of the pegs and sliding it onto another peg, on top of the other disks that may already be present on that peg.
3. No disk may be placed on top of a smaller disk.

As an example, the figure below shows a legal state of the Towers of Hanoi with 4 disks. From this state it is possible to move the smallest disk to the first or second peg.

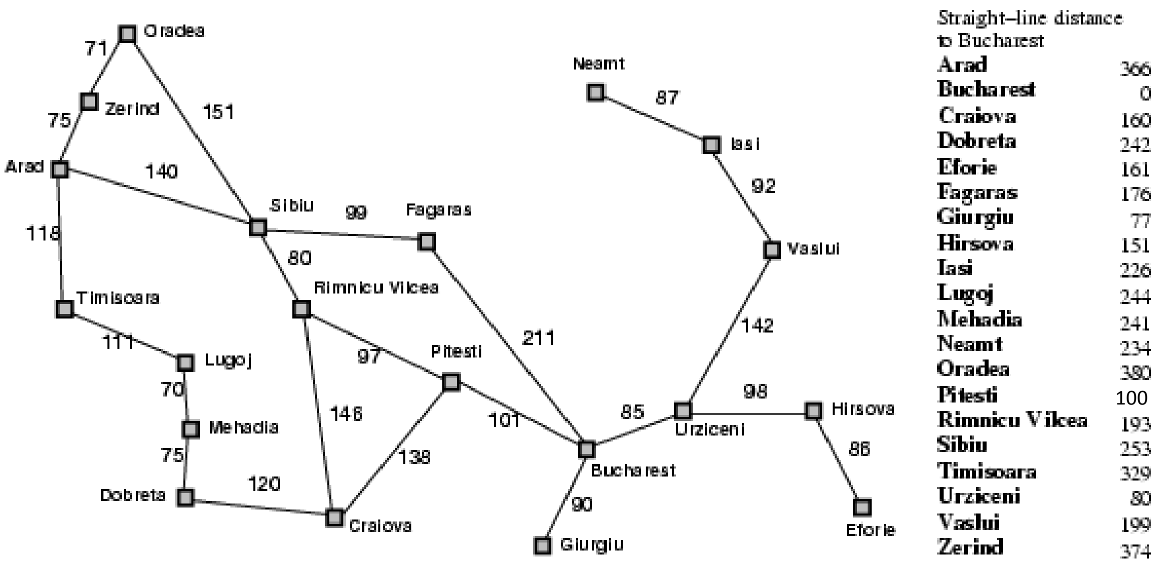


a) Define the Towers of Hanoi as a search problem. That is, define a representation of states and actions, and define
   - the initial state,
   - the actions function,
   - the result function
   - the goal test, and
   - the path cost (assuming that solutions with minimum steps are optimal).

b) Using your state-space definition from a), draw the state-space graph of Towers of Hanoi with *n*=2. Recall that vertices in this graph are legal states and edges are actions causing a change from source to target state.

c) If we assume that all legal configurations of the discs are reachable, what is then the size of the state space as a function of *n*? (This question may vary in difficulty depending on your representation. If you can't find an exact number then try to find a good upper bound)

Exercise 2



Consider the graph-search version of A* using the straight-line distance heuristic $h_{SLD}(n)$ to find a shortest path from Lugoj (L) to Bucharest (B).

a) Finish the table below showing a possible content of the fringe queue in each iteration (in iteration 0, no nodes have been expanded; in iteration 1, one node has been expanded, etc.). Represent a fringe node by a triple <*f*,*g*,*h*,*s*>, where the state *s* is given by the first letter of the city it represents.

Iteration	***Fringe***
0	<244,0,244,L>
1	<311,70,241,M> <440,111,329,T>
…	…

b) What solution is returned by A* after finishing the last iteration in your table?

Exercise 3 (adapted from RN10 3.25)

Let WEIGHTED A* denote the tree-search version of A* using the evaluation function $f(n) = (1\text{-}w)g(n) + wh(n)$, where $0 \leq w \leq 1$ and $h(n)$ is admissible.

a) Which algorithm does WEIGHTED A* correspond to with $w = 1$ and $w = 0.5$, respectively? (w = 0 is also considered in the solutions, but falls outside our curriculum)

b) For which values of w are WEIGHTED A* optimal (Assuming that A* only is optimal if it uses an admissible heuristic)?

Exercise 4 (challenge)

Let h be a consistent heuristic function.

a) Assume that the goal is reachable from n and let $p^*(n)$ denote an optimal solution path from n to a goal state. Prove that $h(n) \leq h^*(n)$ by induction on the number of edges in $p^*(n)$.

b) Use the result in a) to prove that h is admissible.