Introduction to Intelligent Systems: Homework 3

Alvin Lin - Section 1

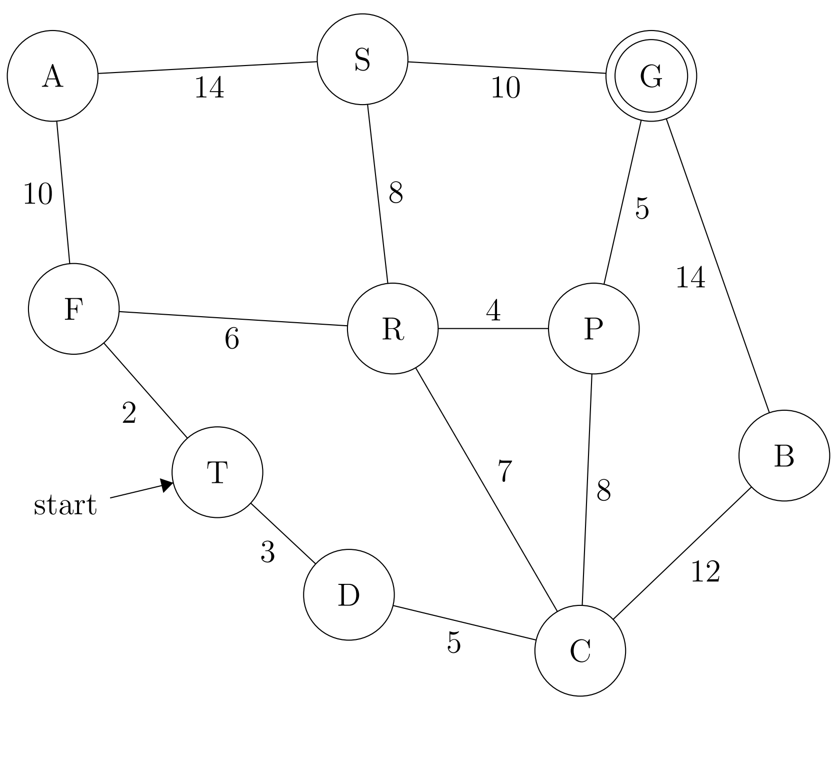
August 2017 - December 2017

Problem 1

Trace the operation of (a) Greedy Best First Search and (b) A* to the problem of getting from node T to node G below using the heuristic of straight-line distance. Show the sequence of nodes that the algorithms will consider and the f,g,h values for each node. For paths that would result in loops, only show the repeated node, do not expand its children. (c) You may have noted that A* seems to return a sub-optimal path. Why is that?

$$h_{SLD}:$$

$$A = 20 \quad B = 10 \quad C = 12$$
$$D = 13 \quad F = 25 \quad P = 4$$
$$R = 10 \quad S = 8 \quad T = 22$$



(a) Greedy Best First Search

Path	Current Node	Neighbors	Choice
$\emptyset$	T	D; $f(D) = h(D) = 13$ F; $f(F) = h(F) = 25$	D
T	D	T; $f(T) = h(T) = 22$ C; $f(C) = h(C) = 12$	C
T,D	C	D; $f(D) = h(D) = 13$ R; $f(R) = h(R) = 10$ P; $f(P) = h(P) = 4$ B; $f(B) = h(B) = 10$	P
T,D,C	P	C; $f(C) = h(C) = 12$ R; $f(R) = h(R) = 10$ G; $f(G) = h(G) = 0$	G

Path: T,D,C,P,G

(b) A* Search

Path	Current Node	Neighbors	Choice
$\emptyset$	T	D; $g(D) = 3; h(D) = 13; f(D) = 16$ F; $g(F) = 2; h(F) = 25; f(F) = 27$	D
T	D	T; $g(T) = 6; h(T) = 22; f(T) = 28$ C; $g(C) = 8; h(C) = 12; f(C) = 20$	C
T,D	C	D; $g(D) = 13; h(D) = 13; f(D) = 26$ R; $g(R) = 15; h(R) = 10; f(R) = 25$ P; $g(P) = 16; h(P) = 4; f(P) = 20$ B; $g(B) = 20; h(B) = 10; f(B) = 30$	P
T,C,D	P	C; $g(C) = 24; h(C) = 12; f(C) = 36$ R; $g(R) = 20; h(R) = 10; f(R) = 30$ G; $g(G) = 21; h(G) = 0; f(G) = 21$	G

Path: T,D,C,P,G

(c) The optimal solution is the path T,F,R,P,G. A* returns a suboptimal path because it factors $h(n)$ too heavily into the heuristic, and $h(n)$ overestimates the cost for some nodes. This is evident since $h(F) > h(D)$ which causes $f(F) > f(D)$ even though F is the node that has the lowest cost to get to the goal.

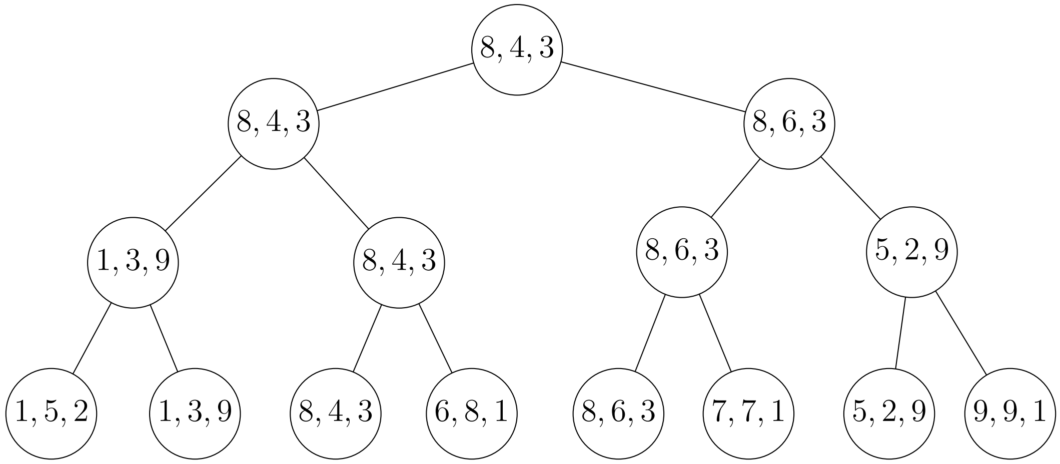
Problem 2

Describe Hill-climbing search. What are some of its limitations?

Hill-climbing search would traverse the nodes and choose a path according to whichever got it closer to the goal (by some heuristic). If the path to the goal requires taking a node that the heuristic considers to be further away, the hill-climbing algorithm would get stuck and never even consider that path.

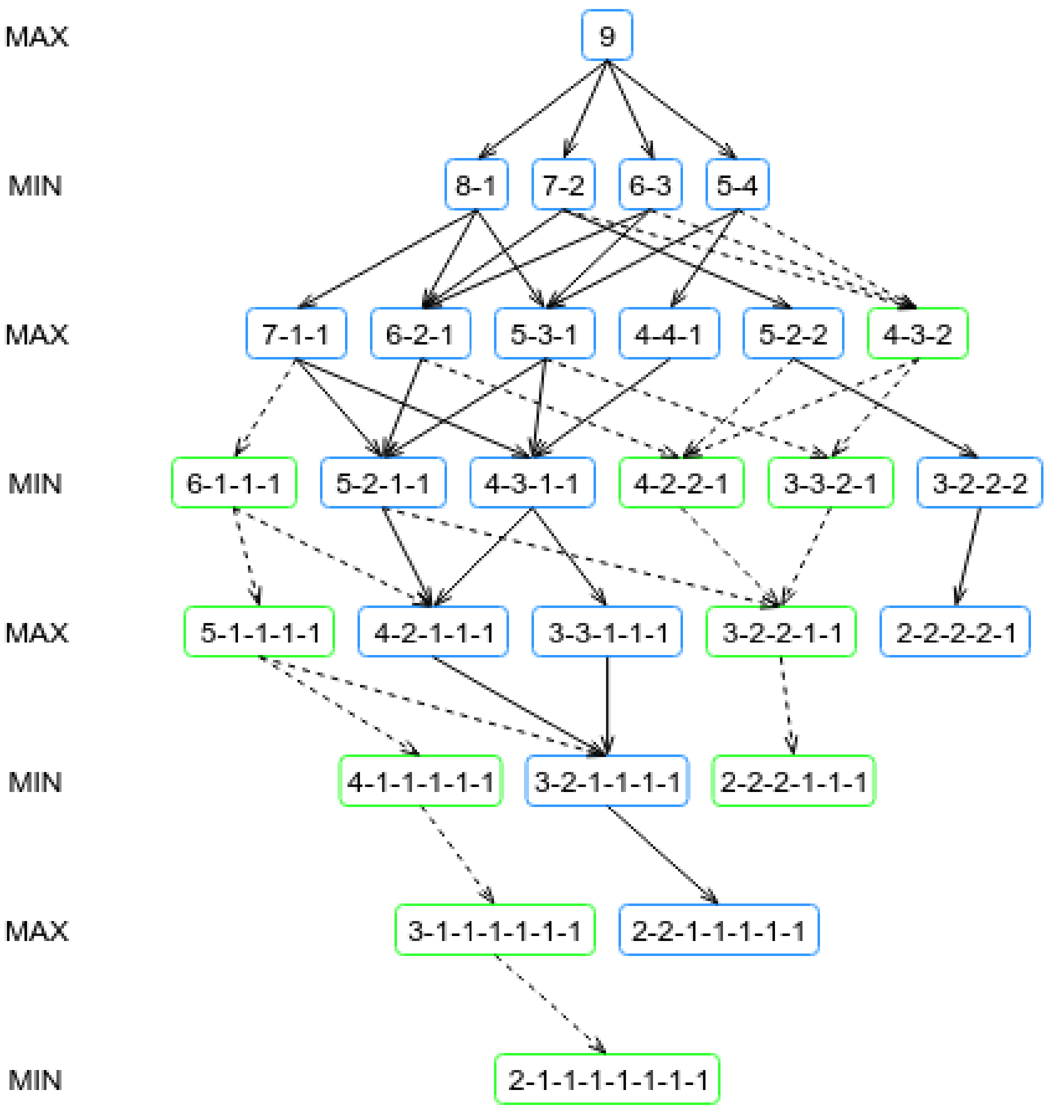
Problem 3

Look at *Figure* 5.4 on Page 166 of the R&N book. Fill in the following **3 player** minimax search tree.



Problem 4

Create and fill-in a Minimax search tree for a 9 token game of Nim. Assume that MAX makes the first move. Fill in the utility value for each node generated.
Nodes colored blue have a utility of 1 and can lead to MIN's victory if played right.



If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech