Introduction to Intelligent Systems: Exam 1

Alvin Lin

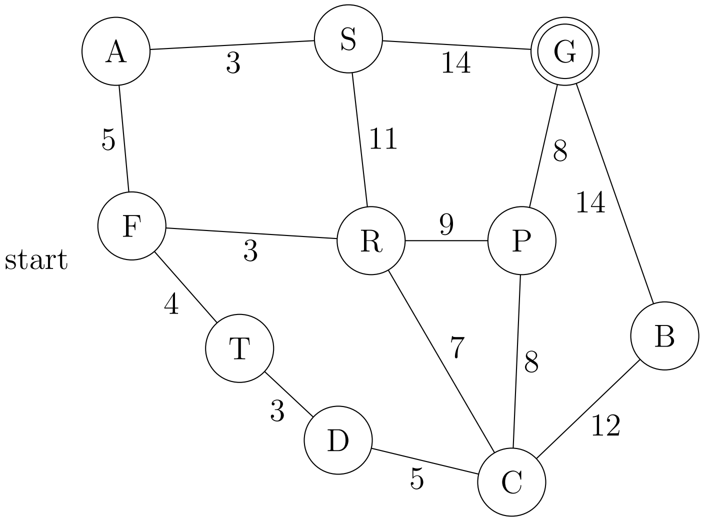
August 2017 - December 2017

Problem 1: A* Search

Trace the operation of A* to the problem of getting from node F 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, and h values for eachnode. Remove paths resulting in loops. When expanding children, list the children in alphabetical order from left to right.

$$h_{SLD}:$$

$$A = 15 \quad B = 10 \quad C = 12$$
$$D = 13 \quad F = 25 \quad P = 9$$
$$R = 10 \quad S = 8 \quad T = 22$$

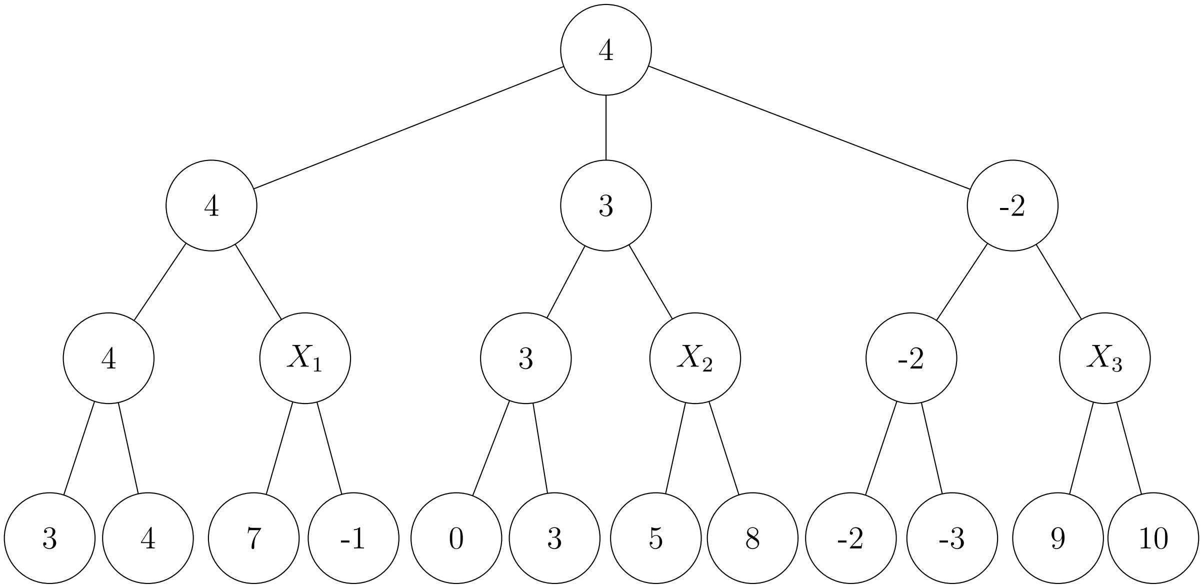


Path	Current Node	Neighbors	Choice
∅	F	A: $g(A) = 5; h(A) = 15; f(A) = 20$ R: $g(R) = 3; h(R) = 10; f(R) = 13$ T: $g(T) = 4; h(T) = 22; f(T) = 26$	R
F	R	C: $g(C) = 10; h(C) = 12; f(C) = 22$ S: $g(S) = 14; h(S) = 8; f(S) = 22$ P: $g(P) = 12; h(P) = 9; f(P) = 21$	P
F,R	P	C: $g(C) = 20; h(C) = 12; f(C) = 32$ G: $g(G) = 20; h(G) = 0; f(G) = 20$	G

Path: F,R,P,G

Problem 2: Minimax with alpha-beta pruning

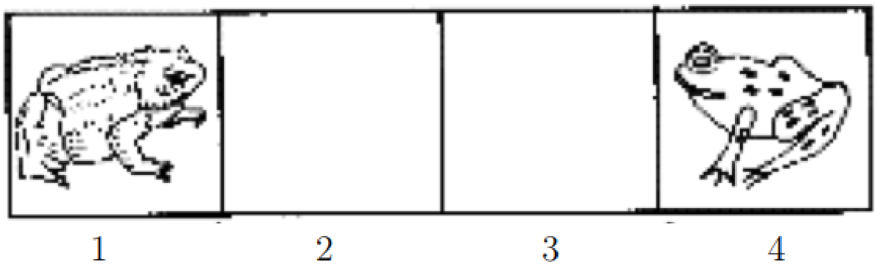
Fill in the following search tree assuming MAX goes first. Put an 'X' over the branch or tree that you can eliminate with alpha-beta pruning.



The branch labeled X_1 can be pruned because a value of 7 was found in the right branch, which is already greater than the max of 4 calculated in the left branch. The branch labeled X_2 can be pruned because a value of 5 was found in the right branch, which is greater than the max of 3 calculated in the left branch. The branch labeled X_3 can be pruned because a value of 9 was found in the right branch, which is greater than the max of -2 calculated in the left branch.

Problem 3: Frogs and Toads

Consider a two-player game featuring a board with four locations, numbered 1 through 4 and arranged in a line. Player A is a Toad and starts on space 1, and player B is a Frog and starts on space 4, as shown in the diagram below. Player A (Toad) moves first.

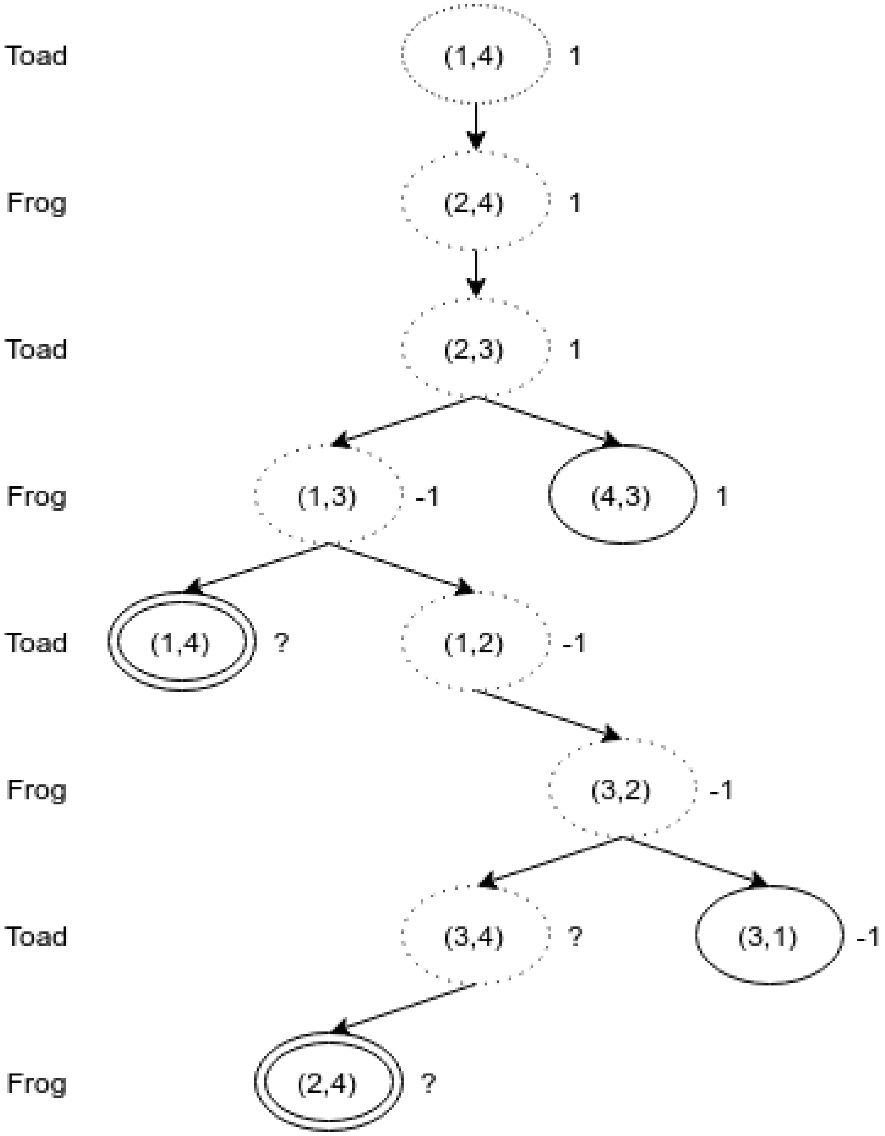


The two players take turns moving, and each player must move to an open adjacent space in *either direction* (no wrap-around) as long as the move is not off the board. If the opponent occupies an adjacent space, then the player may jump over the opponent to the next open space if any. For example, if Toad is on 3 and Frog is on 2, then Toad may move back to 1.

The game ends when one player reaches the opposite end of the board. If player Toad reaches space 4 first, then the value of the game is +1 (a win for Toad); if player Frog reaches space 1 first, then the value of the game is −1 (a win for Frog).

Draw the complete game tree showing only legal moves. If a move results in a loop (repeated state), then show the repeated state but do not expand it further. Assume that Toad moves first and use the following conventions:

1. Write each state as (s_{toad}, s_{frog}) where s_{toad} and s_{frog} denote the token locations. For example the start state is $(1, 4)$.
2. Circle the terminal states (the leaves) and annotate each with its game value.
3. Double-circle the loop states. Since it is not clear how to assign values to loop states, annotate each with a '?'. Do not further expand a loop state.
4. Mark each node with the backed-up minimax value.



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