UNIVERSITY OF RUTGERS

ASSIGNMENT 1: FAST TRAJECTORY PLANNING

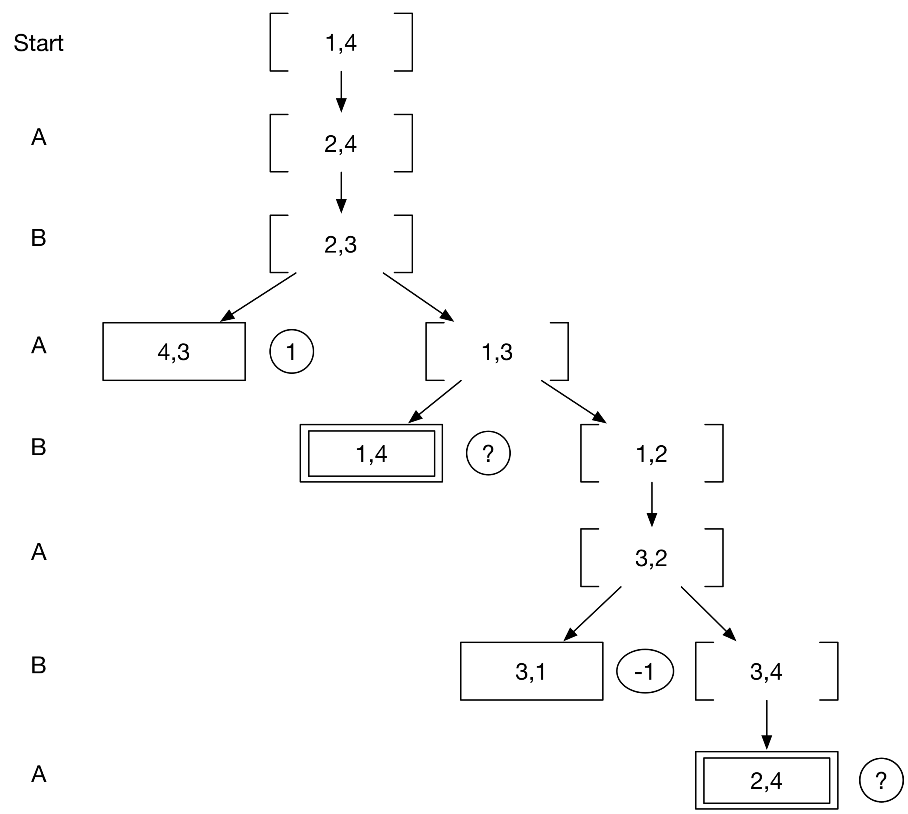
Author:
Xuan WANG
Xunjie ZHU

Supervisor:
Abdeslam BOULARIAS

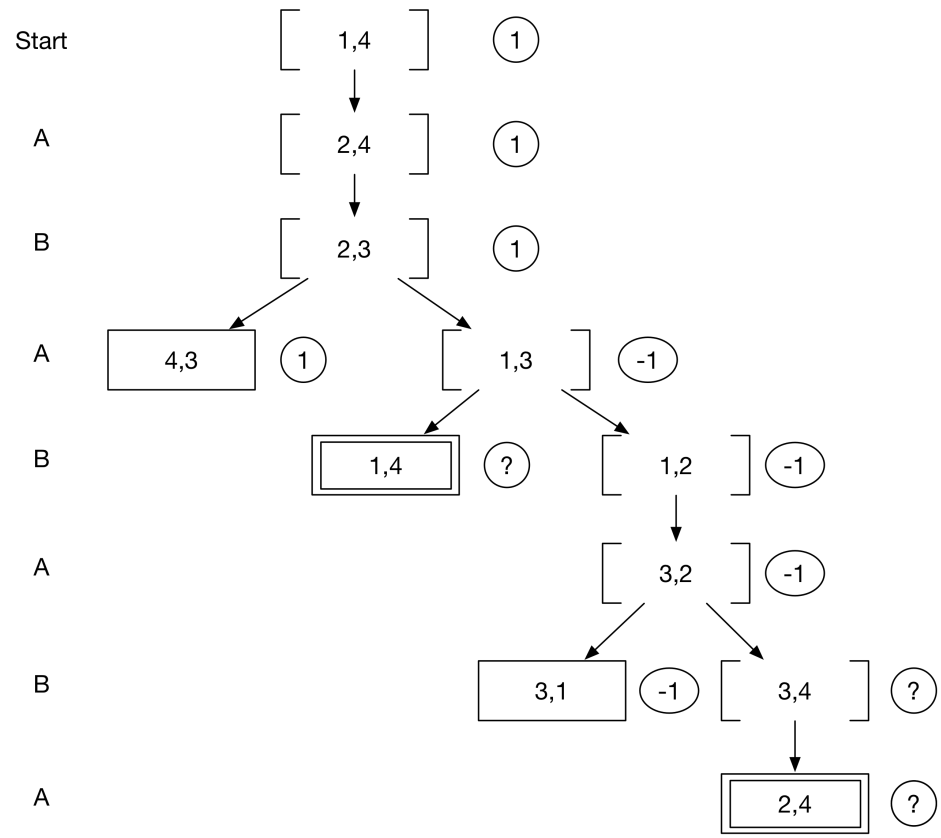
March 6, 2017

PROBLEM 4

(a)Draw the complete game tree



(b)Marked game tree



Decision of B: min(-1,?)=-1, min(?,?)=?

Decision of A: max(1,?)=1, max(?,?)=?

Because in order to win the game, A prefer choosing '1' than '?', B prefer choosing '-1' than '?' as a backed-up value. If two states are both '?', '?' is returned as a backed-up value.

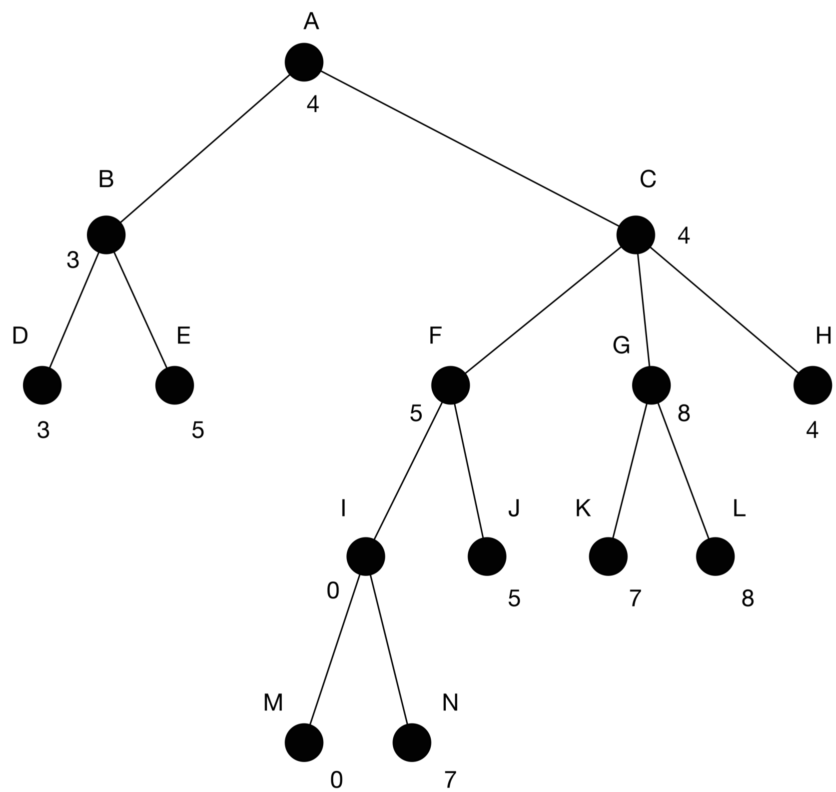
(c)Minimax algorithm failed on this game tree because it runs DFS and meets infinite loop which causes decision tree inifinite. Terminate at the node which has already existed in the tree once, mark it as a loop state and return value '?', handle '?' value with the method mentioned in (b).

If the loop times influences game's value, loop states cannot be denoted as '?', it is a variable and this modified algorithm cannot work at all.

(d)The optimal decision for A and B is constantly going forward, the one that jumps over the other one will win because it moves one more step, so if n is even when A and B meet, A will go across B and A wins, if n is odd, B will go across A and B wins.

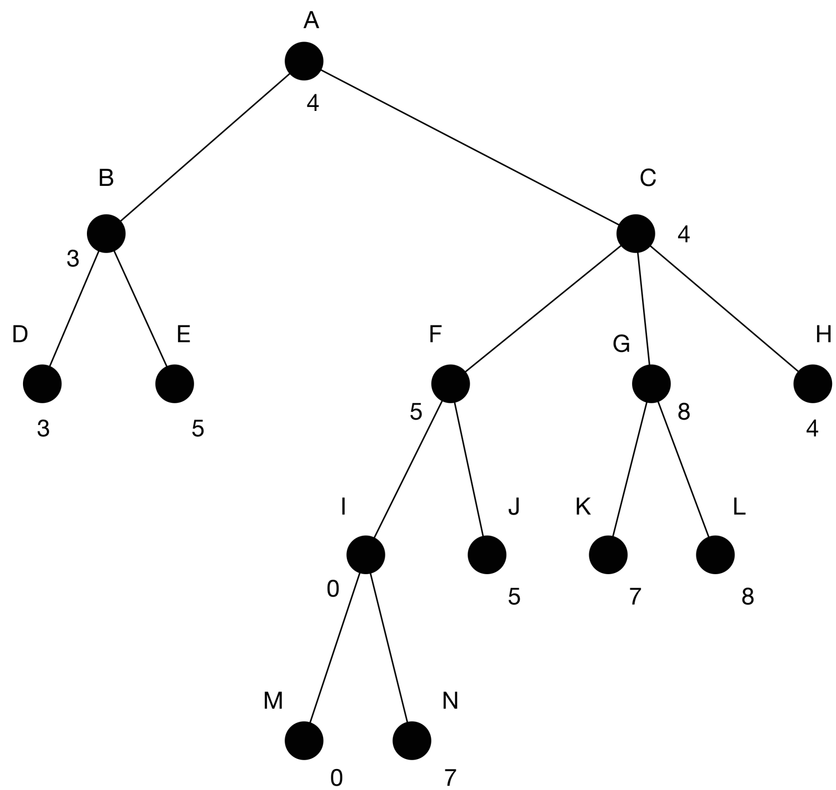
Problem 5

(a)Mark all the nodes

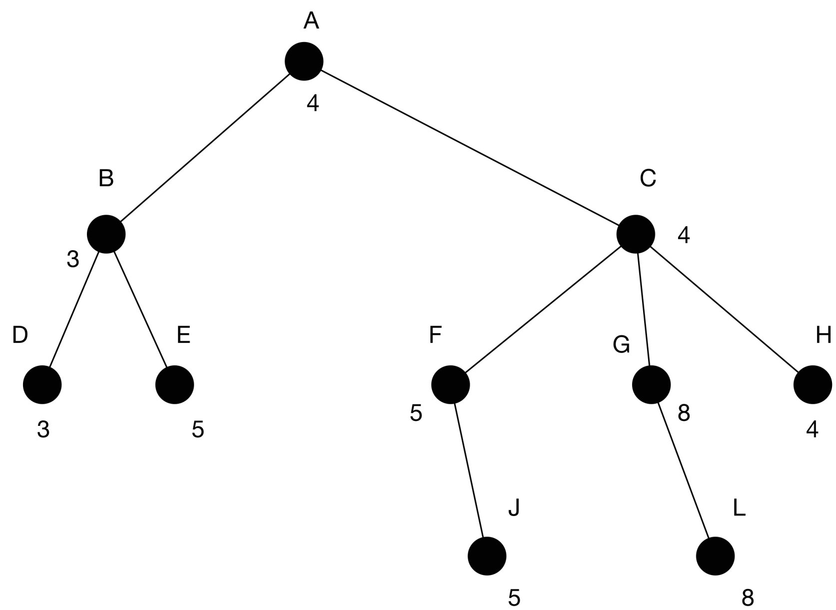


The best move A $\rightarrow$ C, F $\rightarrow$ J, G $\rightarrow$ L.

(2)Nothing changed



(3)Right-to-left



It is the nodes order that determined search times, When nodes are ordered at random, the average number of nodes evaluated is roughly $O(b^{3d/4})$, If the move ordering for the search is optimal (meaning the best moves are always searched first), the number of leaf node positions evaluated is about $O(b^{d/2}) = O(\sqrt{b^d})$. For example, F $\rightarrow$ J will be searched first if it is right-to-left order and F $\rightarrow$ I will be abandoned, but if it is left-to-right order, computer needs to completely search F $\rightarrow$ I first then finds F $\rightarrow$ J.