MIT EECS
6.034: Artificial Intelligence
(Fall 2005)
Instructor: Luis E. Ortiz

Recitation 4: Progressive Deepening for Search in Games (Solution)
October 7, 2005

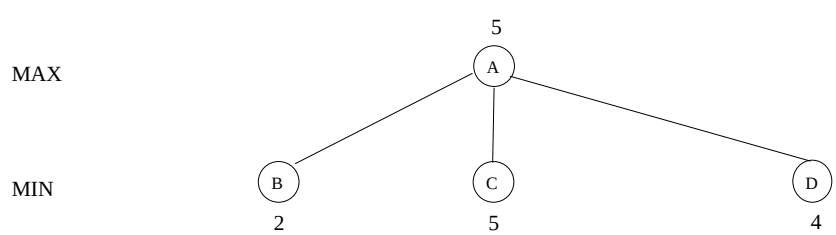
1.

Depth	Best Move	Order Alpha-Beta Applied	Num. Evaluations static	Num. Evaluations nodes
1	C	A B C D	3	4
2	C	A C G H D I B E	4	8
3	B	A C H G D I N O J K P B E L M F	9	16
4	B	A B F E L D I N U O C H	5	12

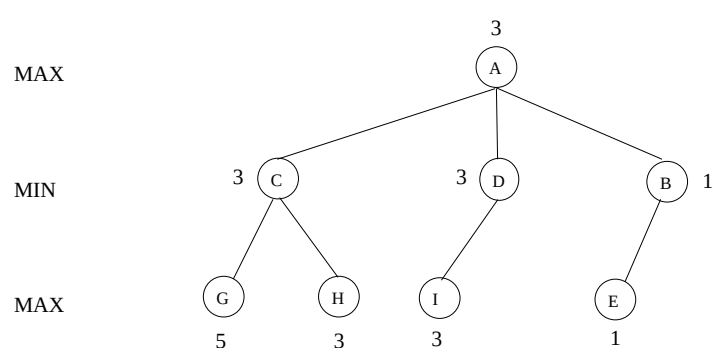
Note how Alpha-Beta did less work at depth 4 than at depth 3, by exploiting information from evaluating up to depth 3 about what the best evaluation order for Alpha-Beta might be (*i.e.*, the node evaluations obtained from Alpha-Beta at depth 3). Using this information does not guarantee an improvement in the performance of Alpha-Beta, but it is very useful as a heuristic.

Below are the trees corresponding to the application of Alpha-Beta at each depth level. The number around a node corresponds to the output of Alpha-Beta after evaluating that node (given also the corresponding values of α and β when Alpha-Beta was called for that node). Only those nodes/states where Alpha-Beta was applied (or, in other words, *visited* by Alpha-Beta) are drawn. The left-to-right order corresponds to the actual order in which Alpha-Beta was applied to the children/successors of the node. Note that, whenever possible, we are using the output evaluations from running Alpha-Beta at the previous depth to order the node evaluations when performing Alpha-Beta up to a particular depth.

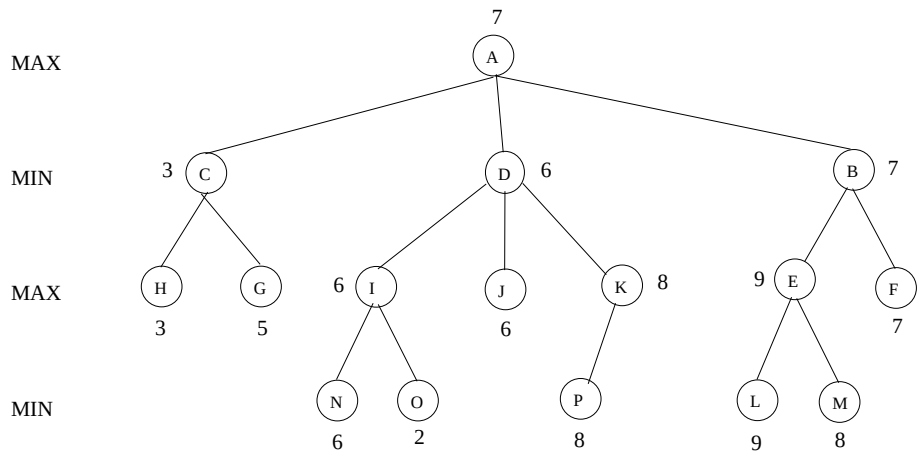
For $d = 1$:



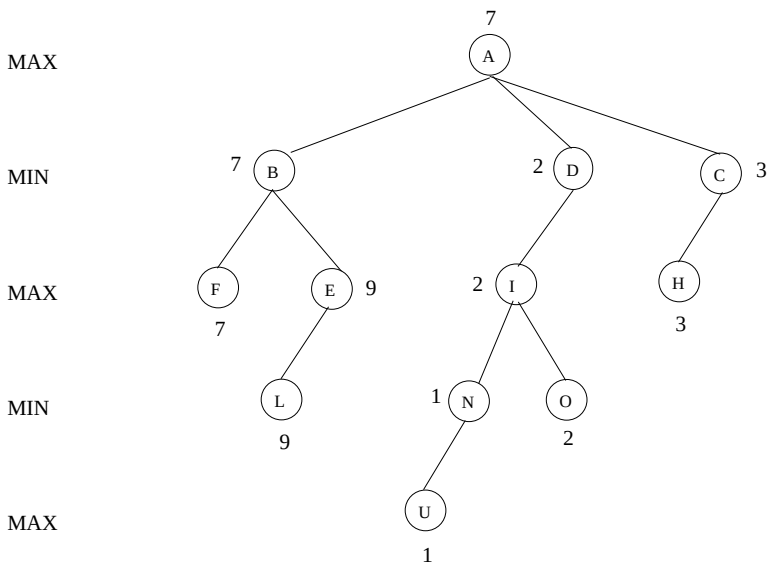
For $d = 2$:



For $d = 3$:



For $d = 4$:



Note that, for each depth, the order in which Alpha-Beta was applied to the nodes is exactly the order in which depth-first search would extend the nodes in the corresponding tree above.