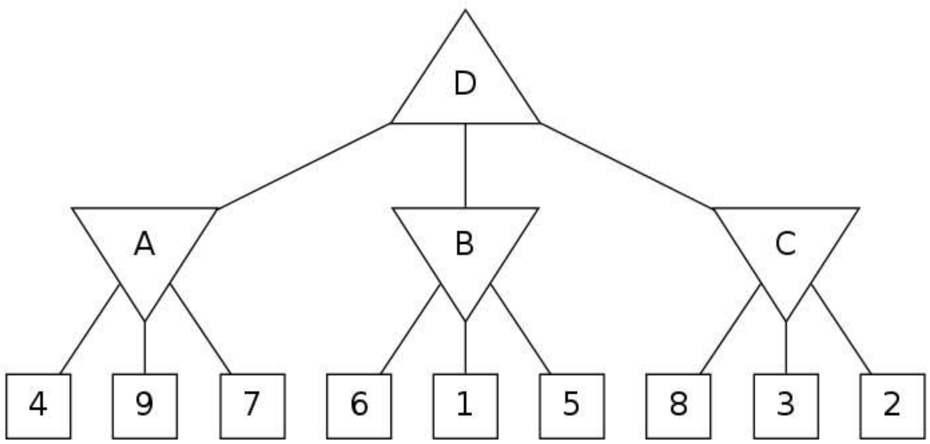
hw3_games_q4_alpha_beta_pruning

Question 4: Alpha-Beta Pruning

0.0/13.0 points (graded)

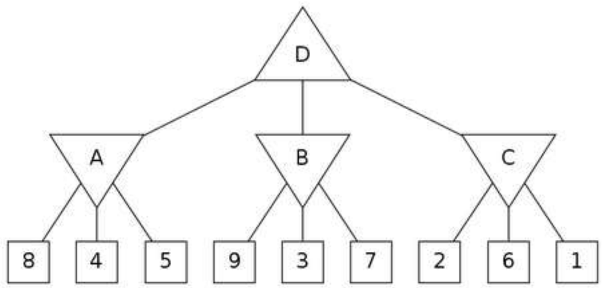
Consider the game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Assuming both players act optimally, use alpha-beta pruning to find the value of the root node. The search goes from left to right; when choosing which child to visit first, choose the left-most unvisited child. In the first set of boxes below, enter the values of the labeled nodes. Then, in the second set of boxes, enter a 'x' for the leaf nodes that don't get visited due to pruning. For leaf nodes that do get visited, leave the corresponding entry blank.

Hint: Note that the value of a node where pruning occurs is not necessarily the maximum or minimum (depending on which node) of its children. When you prune on conditions $V > \beta$ or $V < \alpha$, assume that the value of the node is V.



A	B	C	D
4	1	3	4
Answer: 4	**Answer:** 1	**Answer:** 3	**Answer:** 4

4	9	7	6	1	5	
					x	
					Answer: x	



This problem randomly sets the leaf values for the tree. The solution here is for the above tree, which may be different from the tree in your homework.

A: $\alpha = -\infty$ and $\beta = \infty$. Because $\beta = \infty$, there will be no pruning. Intuitively, this means that any value that the minimizer finds might be used by the maximizer.

B: $\alpha = 4$ and $\beta = \infty$. The first leaf value, 9, is greater than 4, so the other children must be checked. The second leaf value, 3, is less than 4, so the remaining child can be pruned and B gets value 3. Intuitively, B will never take a value greater than 3, and D will never select a value less than 4, so you know that D will never select the middle action.

C: $\alpha = 4$ and $\beta = \infty$. The first leaf value, 2, is less than 4, so the remaining children can be pruned and C gets value 2.

D: Because $\beta = \infty$, none of D's children can be pruned, which is always true for the root. D then takes the max of its children, and has value 4 (which, because none of its children were pruned, is identical to its value when running minimax without pruning)

Submit