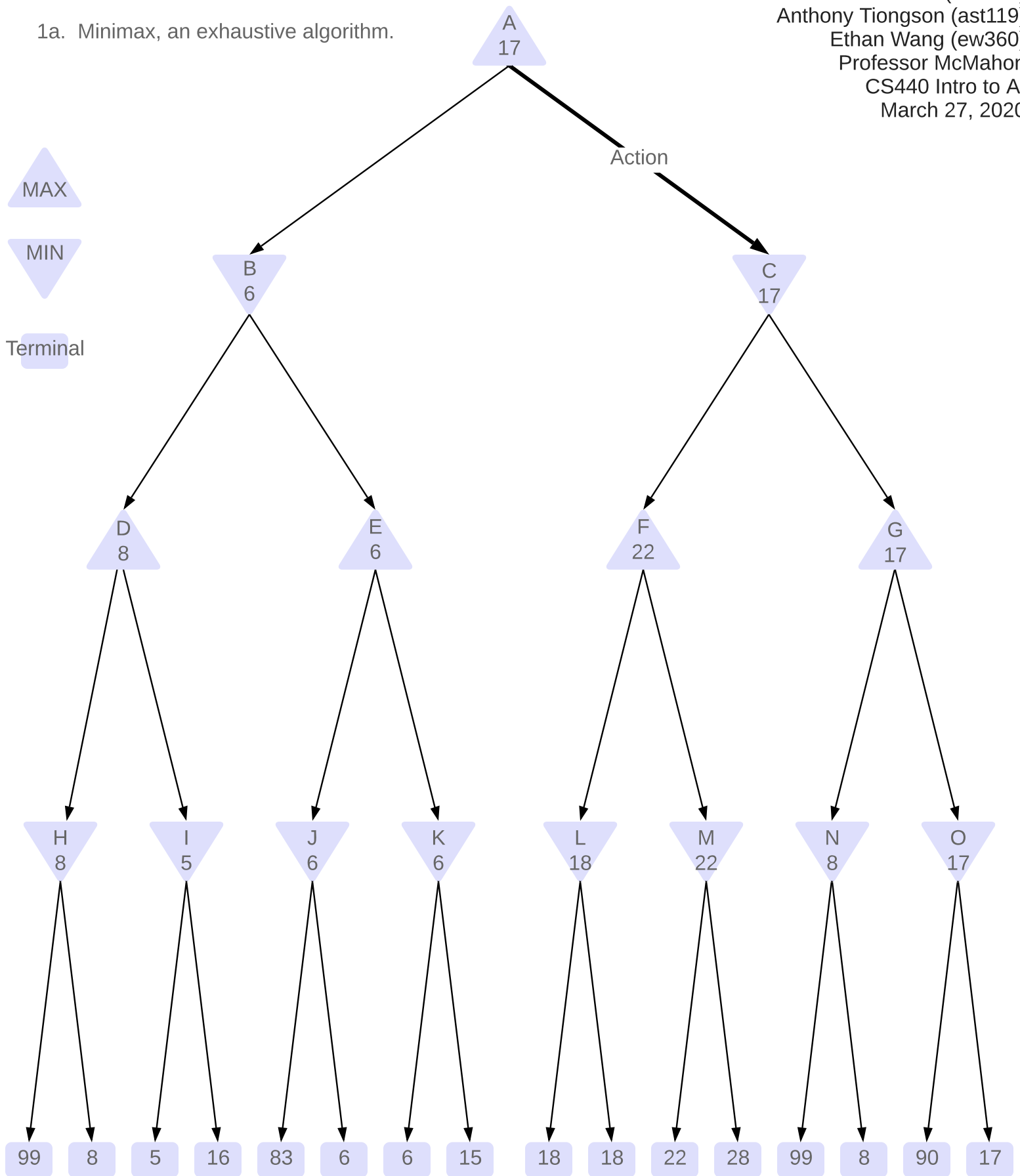


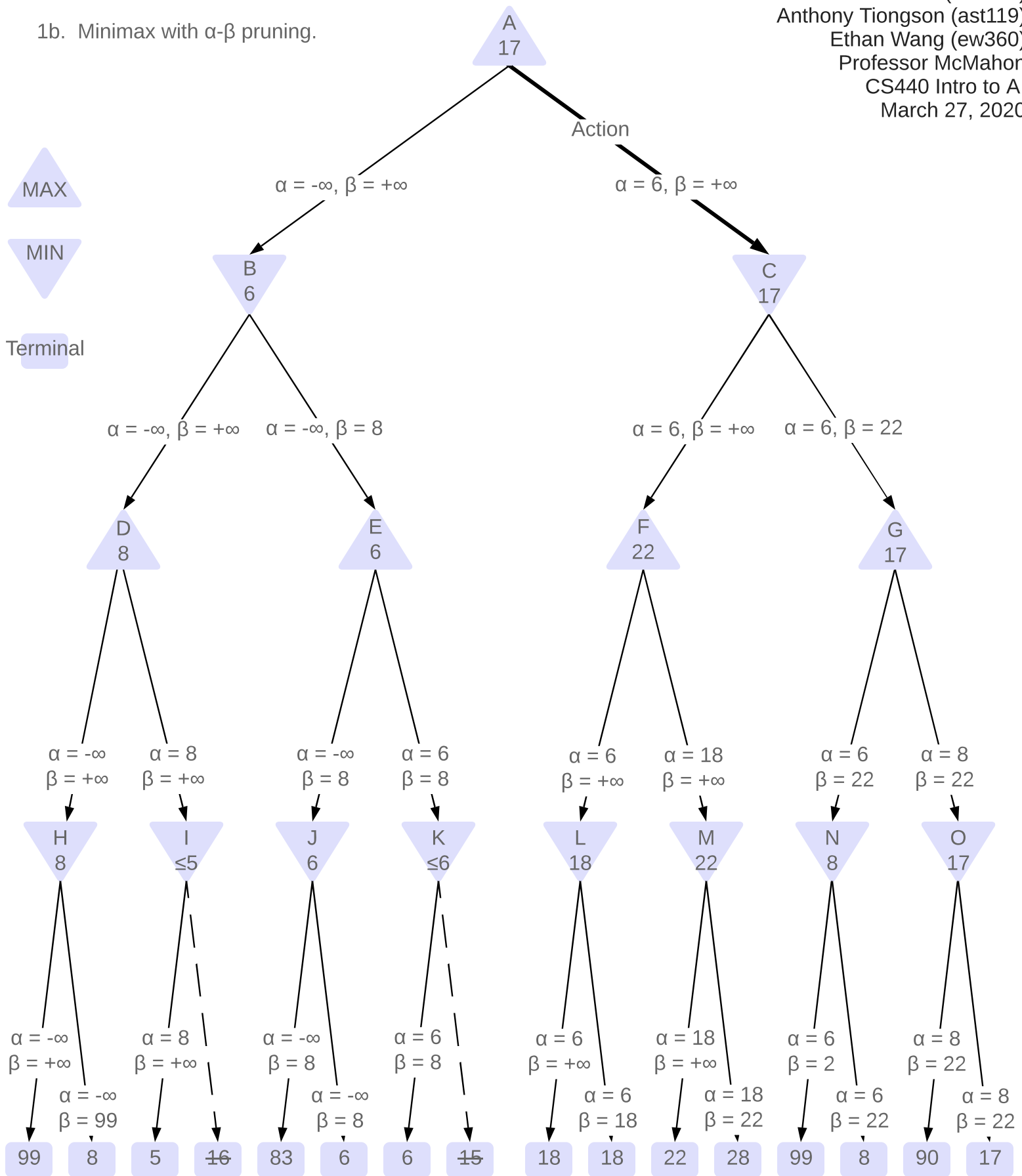
## Assignment 2

Thomas Fiorilla (trf40)  
Srikanth Kundeti (sk1799)  
Anthony Tiongson (ast119)  
Ethan Wang (ew360)  
Professor McMahon  
CS440 Intro to AI  
March 27, 2020

1a. Minimax, an exhaustive algorithm.



1b. Minimax with  $\alpha$ - $\beta$  pruning.



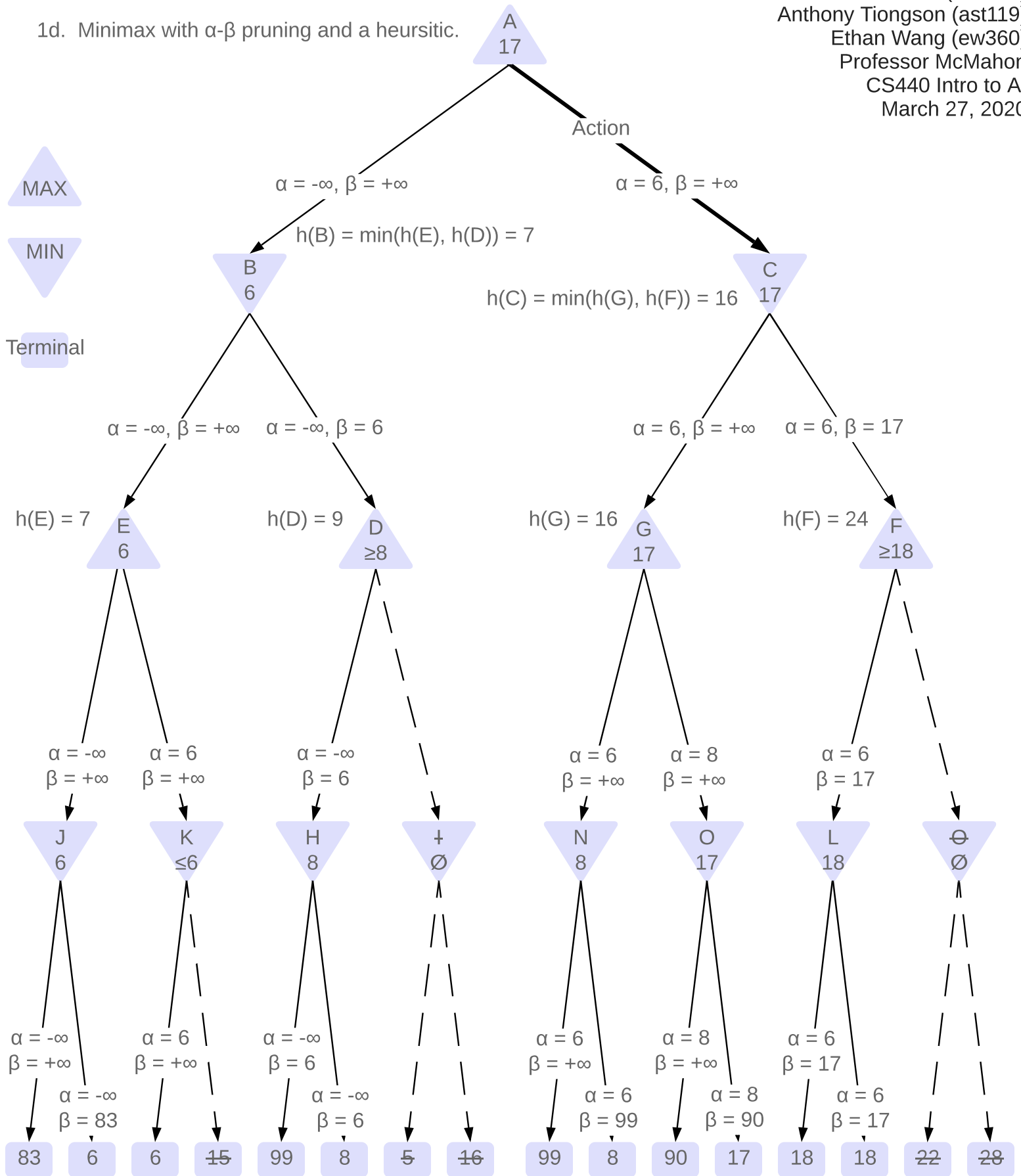
2 nodes pruned.

- 1c. The MAX player at the root state will make the **right** action in the exhaustive minimax algorithm. The MAX player will take the same **right** action when evaluating with  $\alpha$ - $\beta$  pruning. In general, the best move computed by both methods is guaranteed to be the same, which is why  $\alpha$ - $\beta$  pruning is so useful.

# Assignment 2

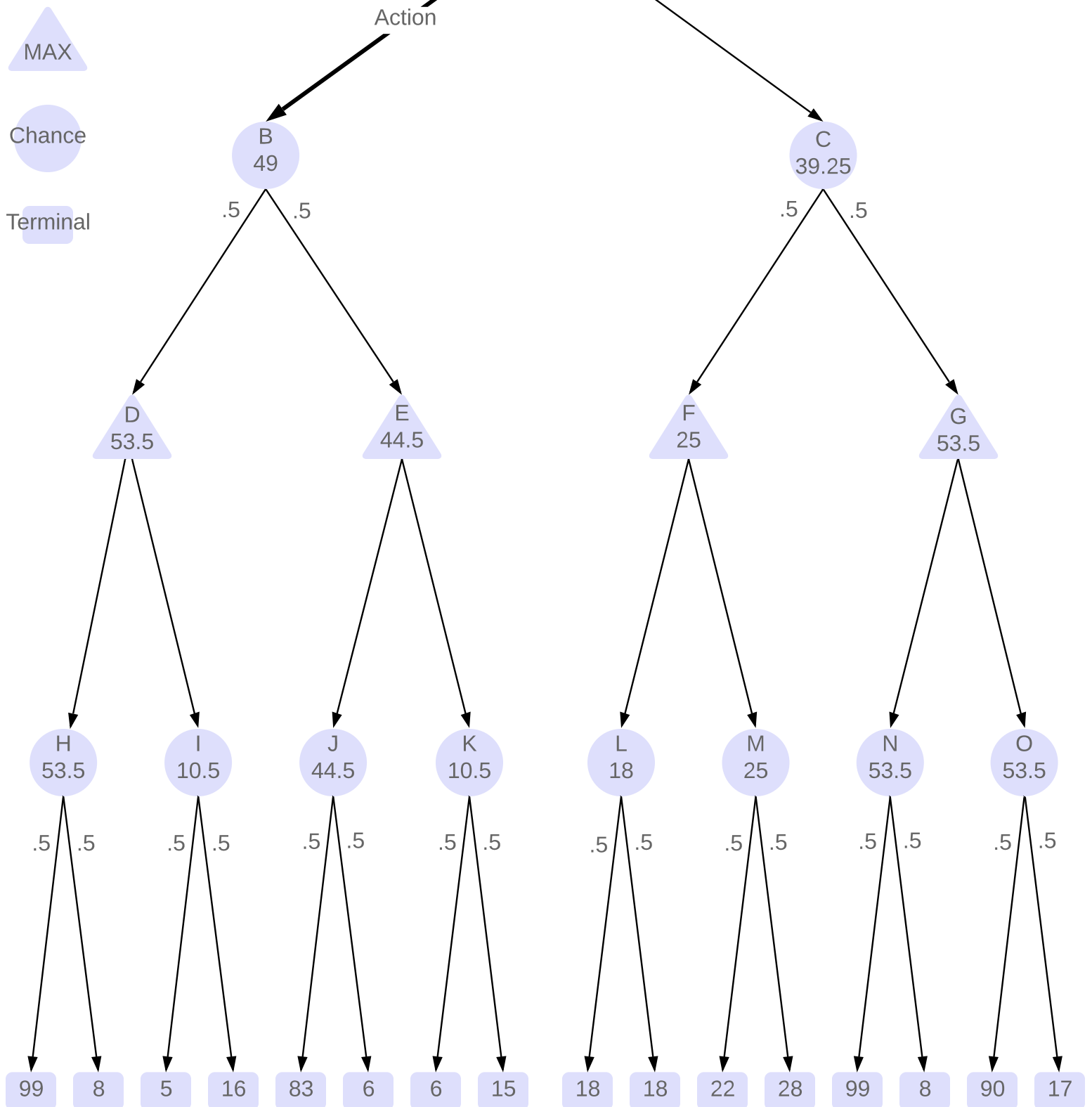
Thomas Fiorilla (trf40)  
Srikanth Kundeti (sk1799)  
Anthony Tiongson (ast119)  
Ethan Wang (ew360)  
Professor McMahon  
CS440 Intro to AI  
March 27, 2020

1d. Minimax with  $\alpha$ - $\beta$  pruning and a heuristic.



7 nodes pruned.

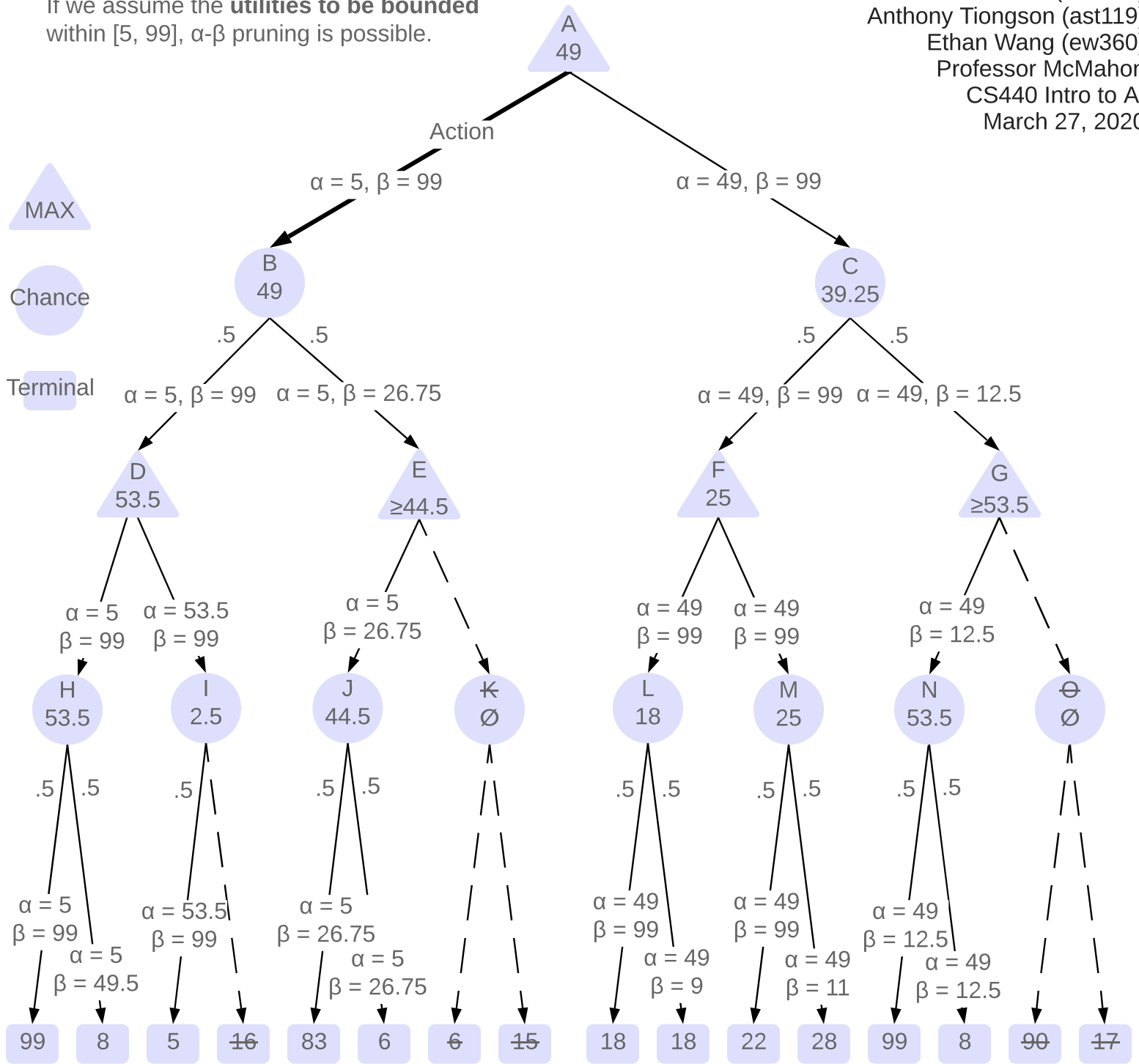
1e. **Expectimax**: the minimizing opponent's moves are governed by a given probability. It behaves and is evaluated more like an environment rather than an adversary. The expected value of these **Chance** nodes is  $.50(\text{left node}) + .50(\text{right node})$ .



**Without bounds** for the utilities, **pruning is not possible** for expectimax so it must be an exhaustive search. In this scenario, any unseen leaf node may possibly be the best or worst value for the tree. Expectimax is an algorithm that is not entirely safe since it has the chance of losing, and ignoring any nodes will greatly decrease its safety and effectiveness.

1e. Expectimax with  $\alpha$ - $\beta$  pruning.

If we assume the **utilities to be bounded** within  $[5, 99]$ ,  $\alpha$ - $\beta$  pruning is possible.



In this scenario, the initialization of the nodes differs to minimax  $\alpha$ - $\beta$  pruning since  $\alpha$ 's lower limit is now 5 and  $\beta$ 's upper limit is 99. Pruning with regards to the MAX nodes is similar to minimax, with the  $\beta$  values now representing the best Chance option in the path to root. The MAX node will additionally compare its current value  $v$  to the max value and prune the next Chance node if so, since the MAX node is guaranteed to select the prior Chance node. Pruning of the Chance nodes will utilize the  $\alpha$  values, but the evaluation is different than the MIN nodes algorithm. For the Chance nodes, the evaluation first checks to see if  $\alpha$  is less than the current Chance node's value  $v$ . If  $\alpha$  is less than  $v$ , the algorithm will continue to the next node to evaluate. If  $\alpha$  is greater than or equal to  $v$ , the algorithm will calculate a value  $x = (\alpha - v)/P(x)$ , where  $P(x)$  is the probability of the next node to evaluate and  $x$  represents the greatest value the next node can be for the Chance node to be as large as  $\alpha$ . If  $x$  is within the bounds of the utility values, the algorithm will continue to the next node for evaluation since there is now a possibility the Chance node can be greater than  $\alpha$ ; if  $x$  is outside the bounds, that next node will be pruned since there is no way for the Chance node to surpass  $\alpha$ .