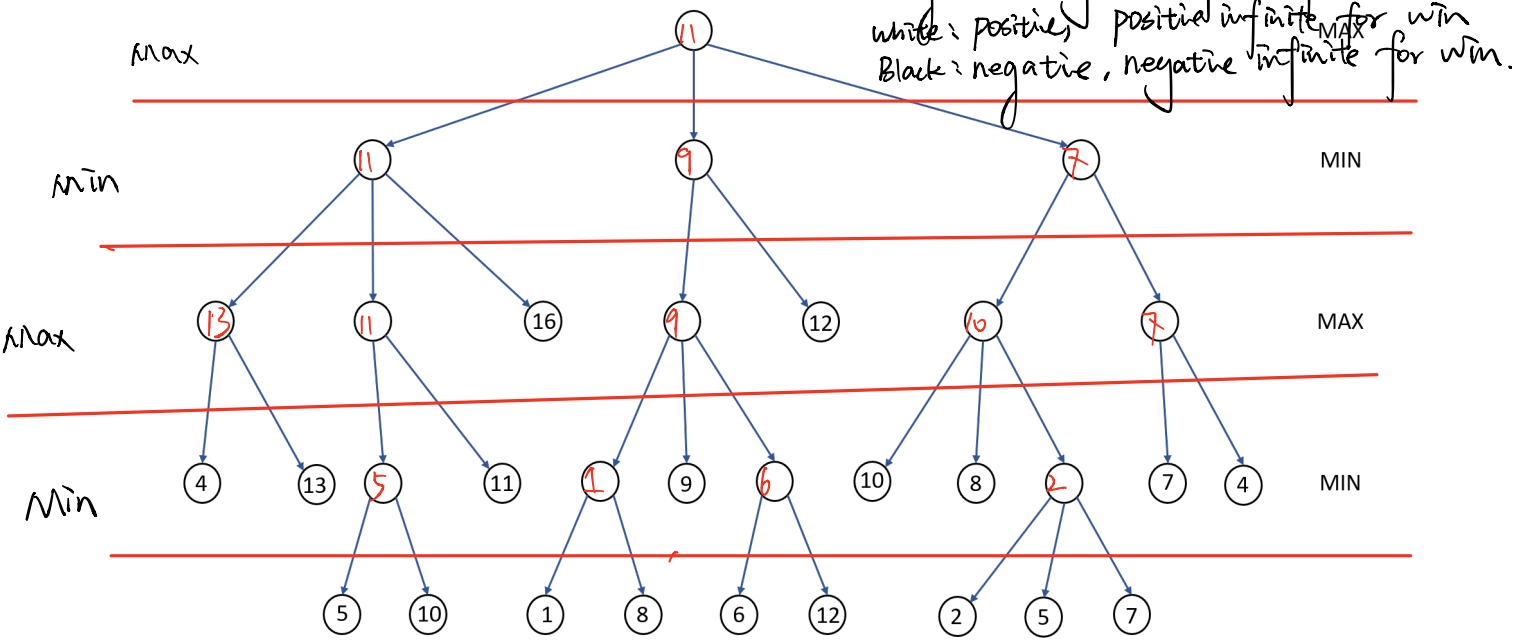
Problem 1: Minimax and Alpha-Beta [15 points]

(a) [4] Use the **Minimax** algorithm to compute the minimax value at each node for the game tree below.

white : positive, positive infinite for win
Black : negative, negative infinite for win.
white : positive, positive infinite for win
Black : negative, negative infinite for win.

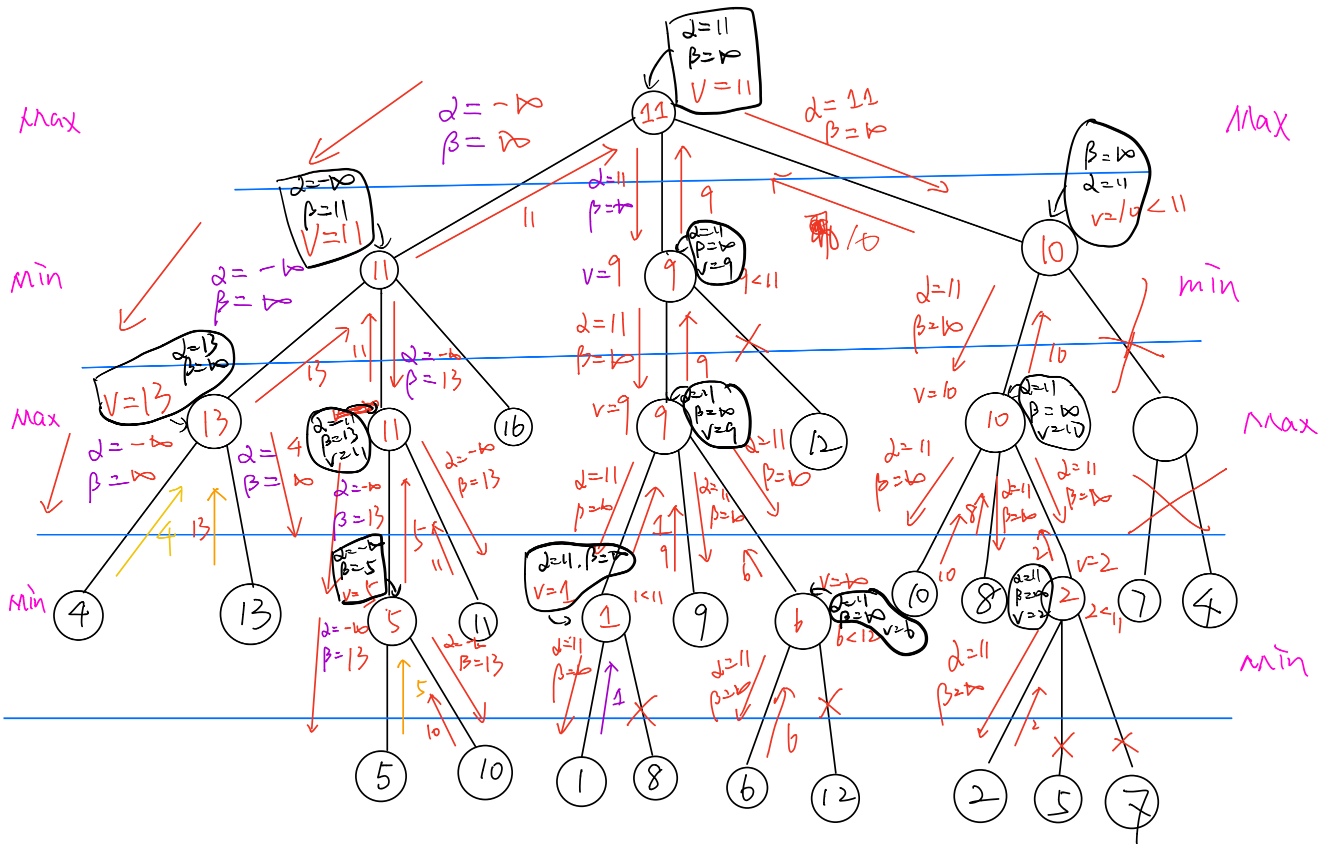


(b) [9] Use the **Alpha-Beta** pruning algorithm to prune the game tree above assuming child nodes are visited from left to right. Show all final alpha and beta values computed at the root, each internal node visited, and at the top of pruned branches. Note: Follow the algorithm in Figure 5.7 in the textbook. Also show the pruned branches.

(c) [2] For a general game tree (i.e., *not* limited to the above tree), are there *any* cases that the Alpha-Beta algorithm gives a *different value at the root node* than the Minimax algorithm? If yes, show an example; if no, just say no.

NO!

part b



①: root initilized to the worst case for the Maximizer. (最大值的最差情况, that is $-\infty$, so it can not bigger than anyone)

②: do a left-right traversal.

③: Find α and β, 那 now there is no explored node, so that, α is $-\infty$ (worst case) β is ∞ (worst case)

④: 到下一个 node, 是 Min, initial to worst case. is ∞.

⑤: 一直往下, 直到 leaf node, then pass value back, 然, update the value that better for nodes, like,

Max: better than worst case $-\infty$. → After update, if current value here is better than β, do not need to worry about futher childre.

Min: better than worst case ∞. → if current value is smaller than α, than we don't need to worry about futher children.

min 和 α 比 — min < α, don't visit other children

Max 和 β 比大 — max > β. don't visit.