Alpha-beta-Pruning

Name:- Shweta. G. Bhoir

Roll no:- 09

Sub:- IS LAB

Class :- BE/IT

DOA	DOC	Remark	Sign

* MinMax algorith with alpha beta pruning.

→

Alpha-beta Pruning :- Alpha beta Pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.

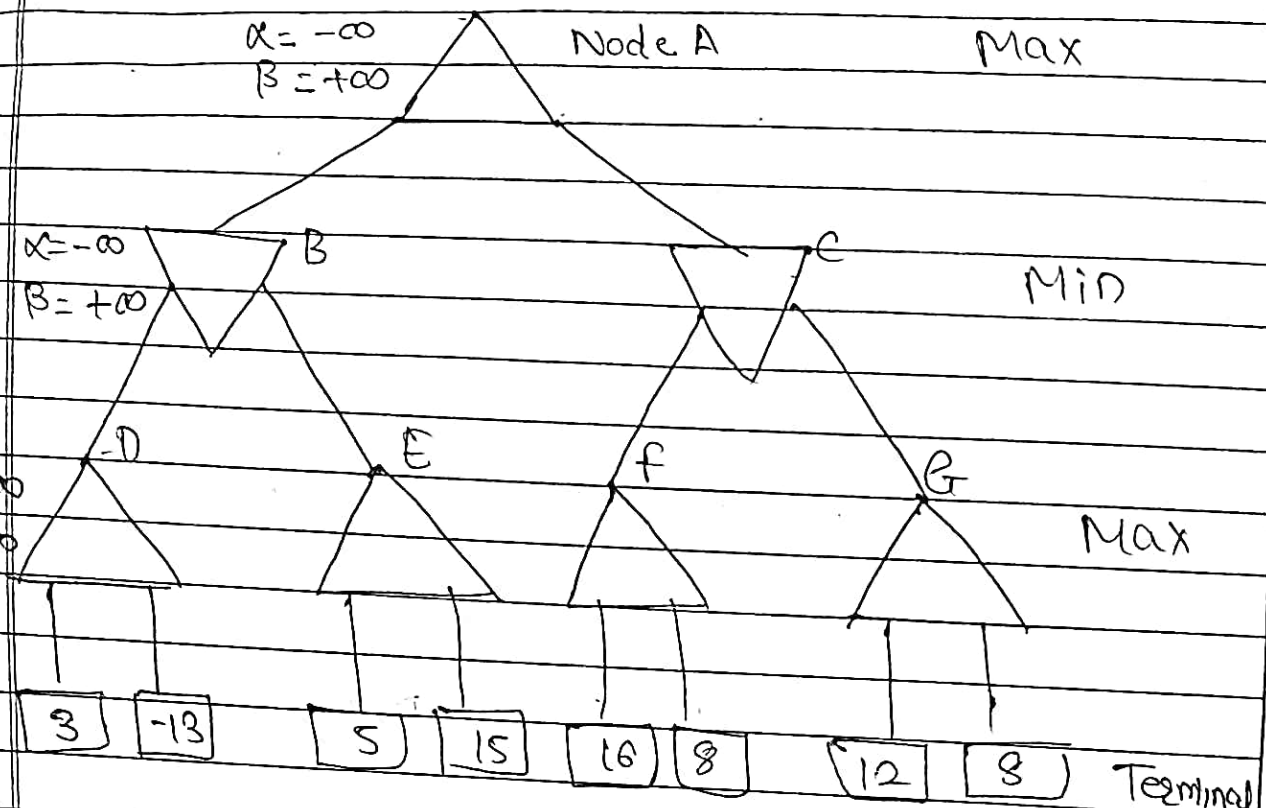
Alpha (α) = The best (highest-value)
= Initial value of alpha is $-\infty$

Beta (β) = The best (lowest-value)
= Initial value of beta is $+\infty$.

The alpha-beta Pruning to a Standard minimax algorithm returns the same move as the Standard algorithm does, but it removes all the nodes as which are not really affecting the final decision but making algorithm slow Hence by pruning these nodes, it makes the algorithm fast.

Rules & Conditions

- The Max player will only update, the value of alpha.
- The min player will only update the value of beta
- We will only pass the alpha, beta values to the child nodes.
- Node values will be passed to upper nodes instead of values of alpha and beta condition to prune = $\alpha \geq \beta$ or $\beta \leq \alpha$.

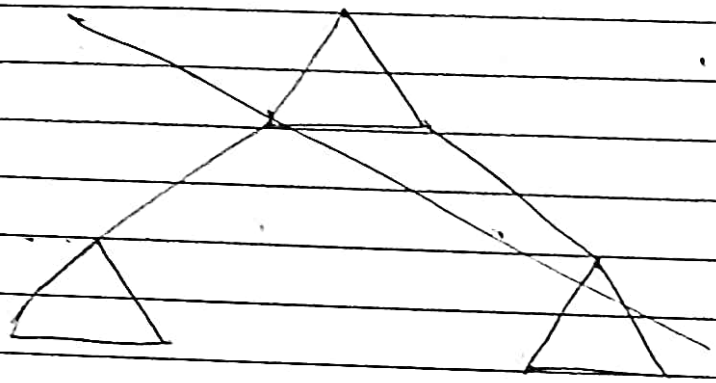


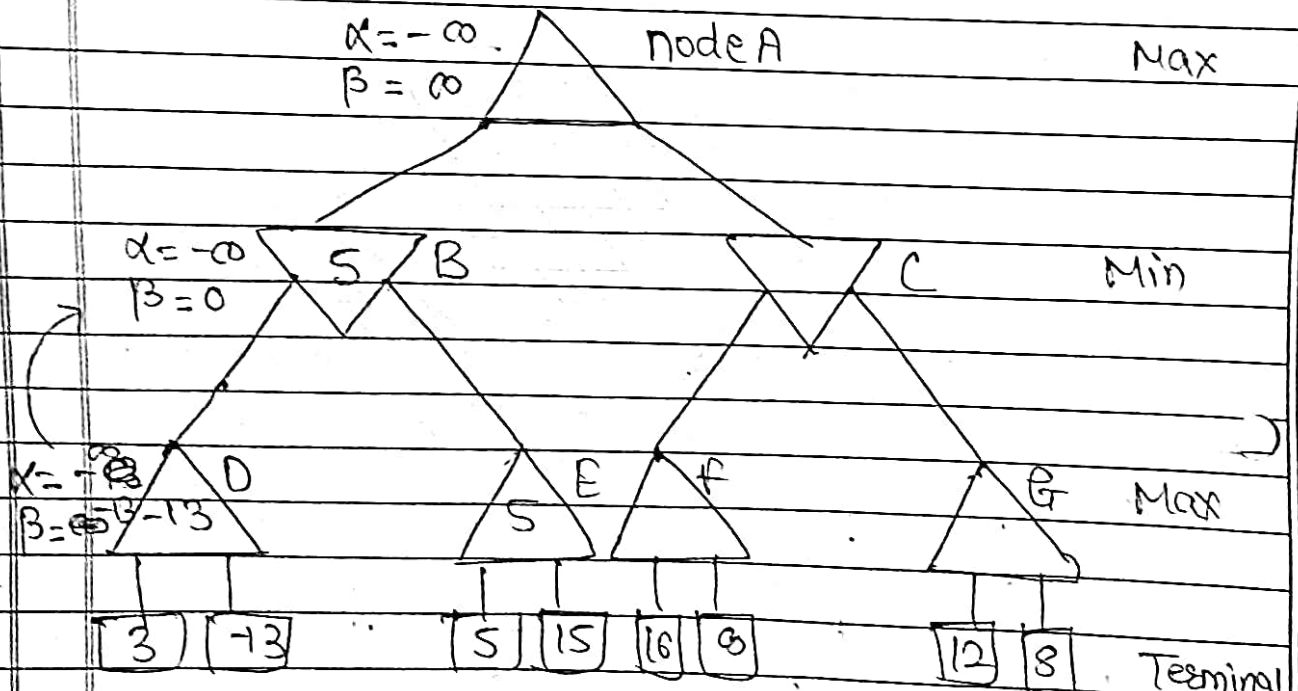
Step 1:- At the first step the, max player will start first move from node A where $\alpha = -\infty$ and $\beta = +\infty$ these values of alpha, beta passed down to node B where again $\alpha = -\infty$ and $\beta = -\infty$ and Node B passes same value to its child D.

Step 2:- At Node D the value of α will be Calculated as its turn for max the value of α compared with firstly 3 and the -13 and max (3, -13) = -13 will be the value of α at node D and node value will be -13

Step 3:- Now algorithm backtracks to nod B where the value of β will changed as this is a turn of min, now $\beta = +\infty$, will compare with the

available subsequent nodes value i.e
min(∞, -13) = -13 hence
at node B now α = -∞ β = -13

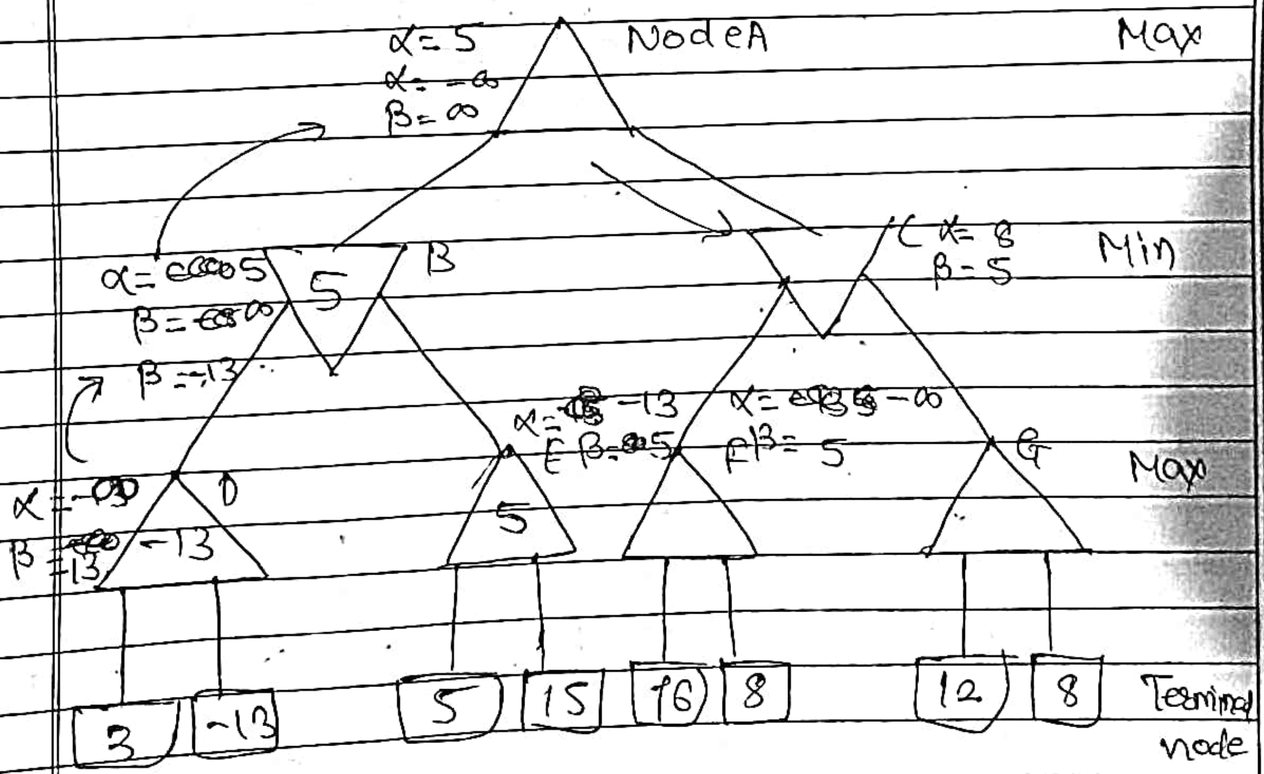




In the next step, algorithm traverse the next successor of node B which is node E and the values of α = -∞ and β = -13 will also be passed.

Step 4 :- At node E, max will take its turned the value of alpha will change the current value of α will be compared with 5 so max$(-\infty, 5) = 5$ hence at node E $\alpha = 5$ and $\beta = -13$ the α will be compared with 15 so max$(5, 15)$ = so the value of $\alpha = 5$ $\beta = -13$ so at node E value will be 5.

For node B its Min turn so the value of β will be changed so early value of β was -13 now $\beta = \min(-13, 5) = 5$ so the value of node will be 5.



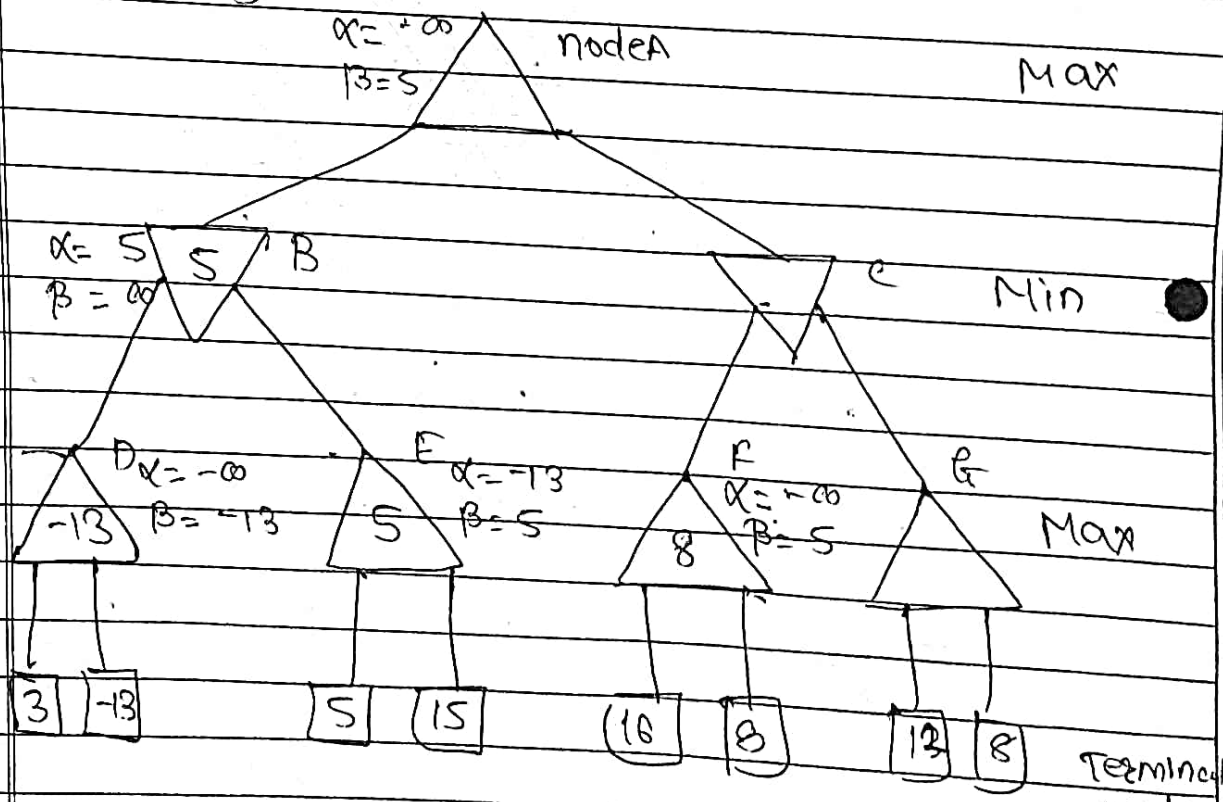
Step 5:- At next step algorithm again backtraces the tree from Node B to Node A The value of α will be changed the max value will be

$\alpha = (-\infty, 5) = 5$

$\beta = \infty$.

These two values now pass at own to the right successor which is node C at the node C $\alpha = 5$ and $\beta = \infty$ the same values will be passed on to node F.

Step 6:- At node F again the value of α will be compared with left child which is 8 and $\max(16, 8) = 8$ so the node value will become 8.

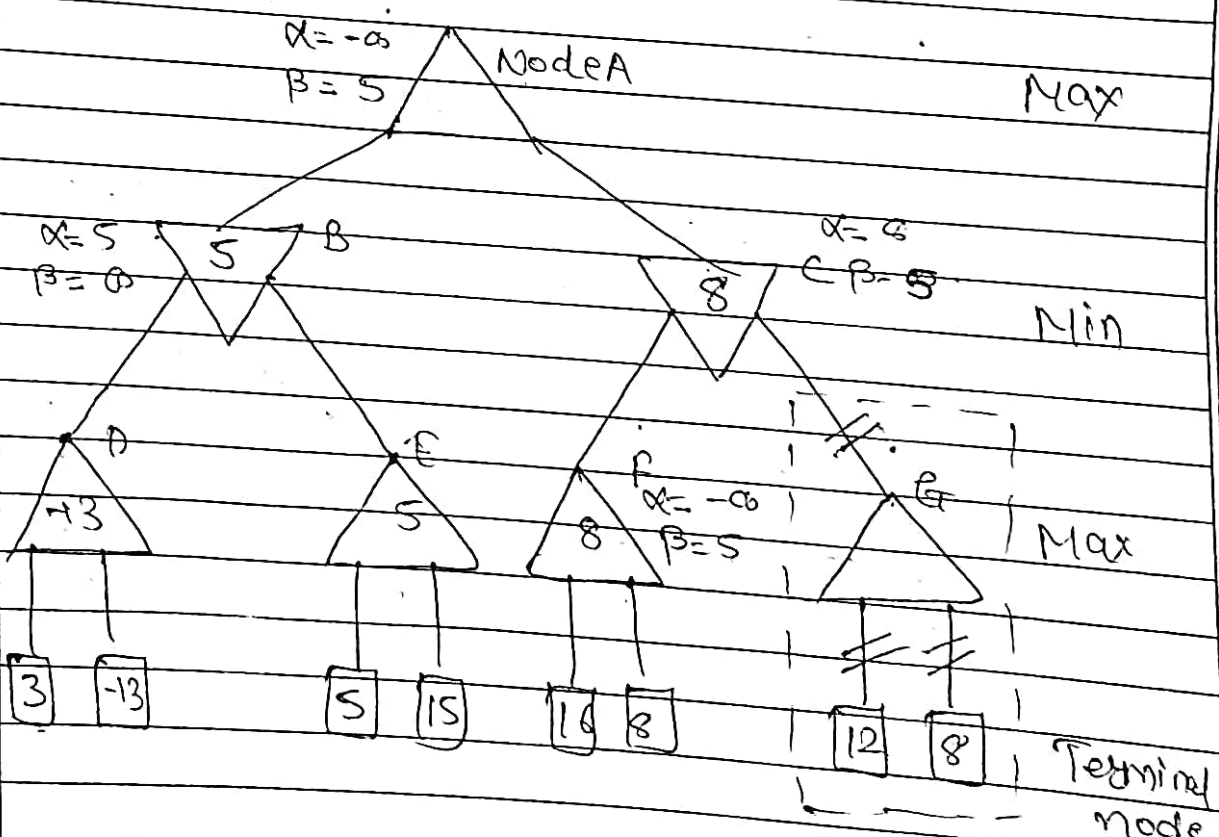


Step 7: At node C $\alpha = 8$ $\beta = \infty$ here the value of β will change it will compare with 8 so now

$\min(\infty, 5) = 5 = \beta$.

So now we have. $\alpha = 8$ and $\beta = 5$ Here the condition to prune i.e $\alpha \geq \beta$ satisfies so the next Right node of the node C will be pruned and the node value of C will become 8.

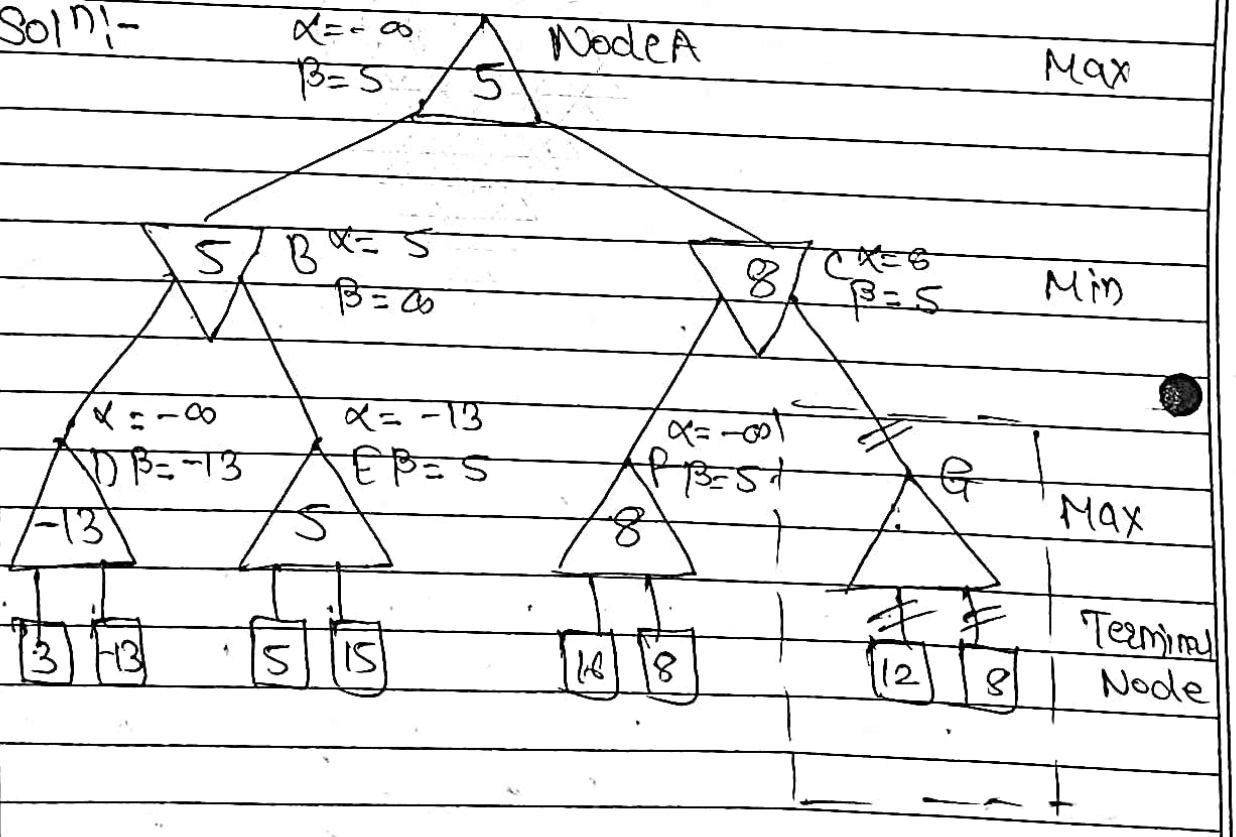
∴ at C $\alpha = 8$ and $\beta = 5$.



Step 3:- Now returns the value of 8 to A be hence the best value of A is
$\alpha = \text{Max}(5, 8) = 5$.
∴ So the final value of node A will be 5
$\alpha = -\infty \quad \beta = 5$ at A.

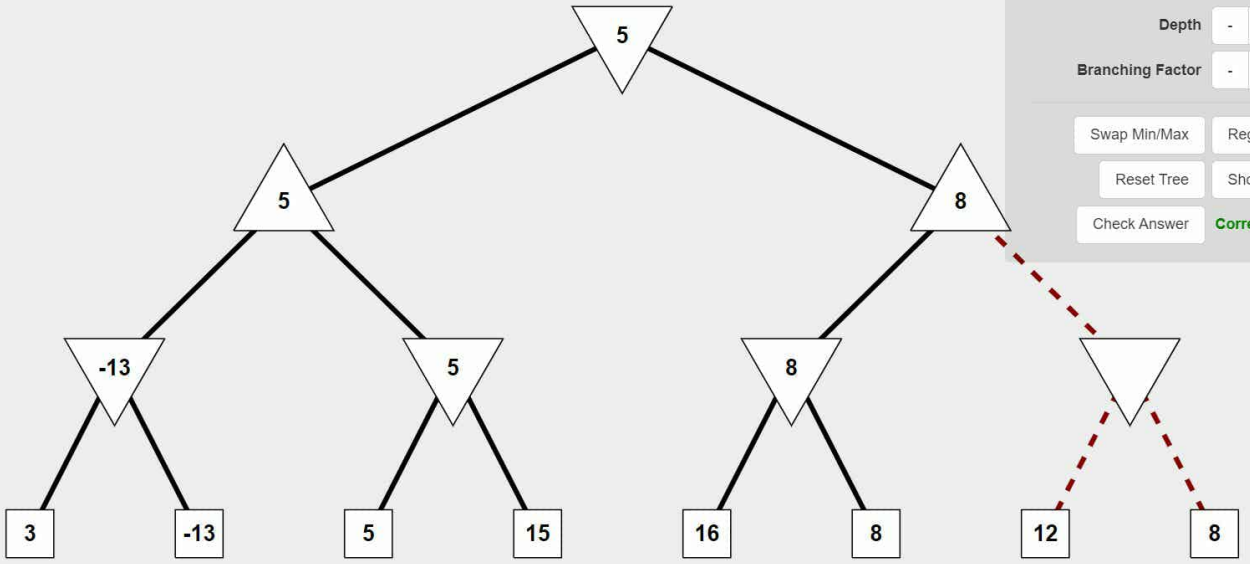
following is the final game tree which is showing the nodes which are compound and nodes which has neve computed Hence the optimal value for the Maximize is 5 for this tree.

Soln:-



Alpha-Beta Pruning Practice

αβ Cutoff



Start Animation

Depth - +

Branching Factor - +

Swap Min/Max | Regenerate Tree

Reset Tree | Show Solution

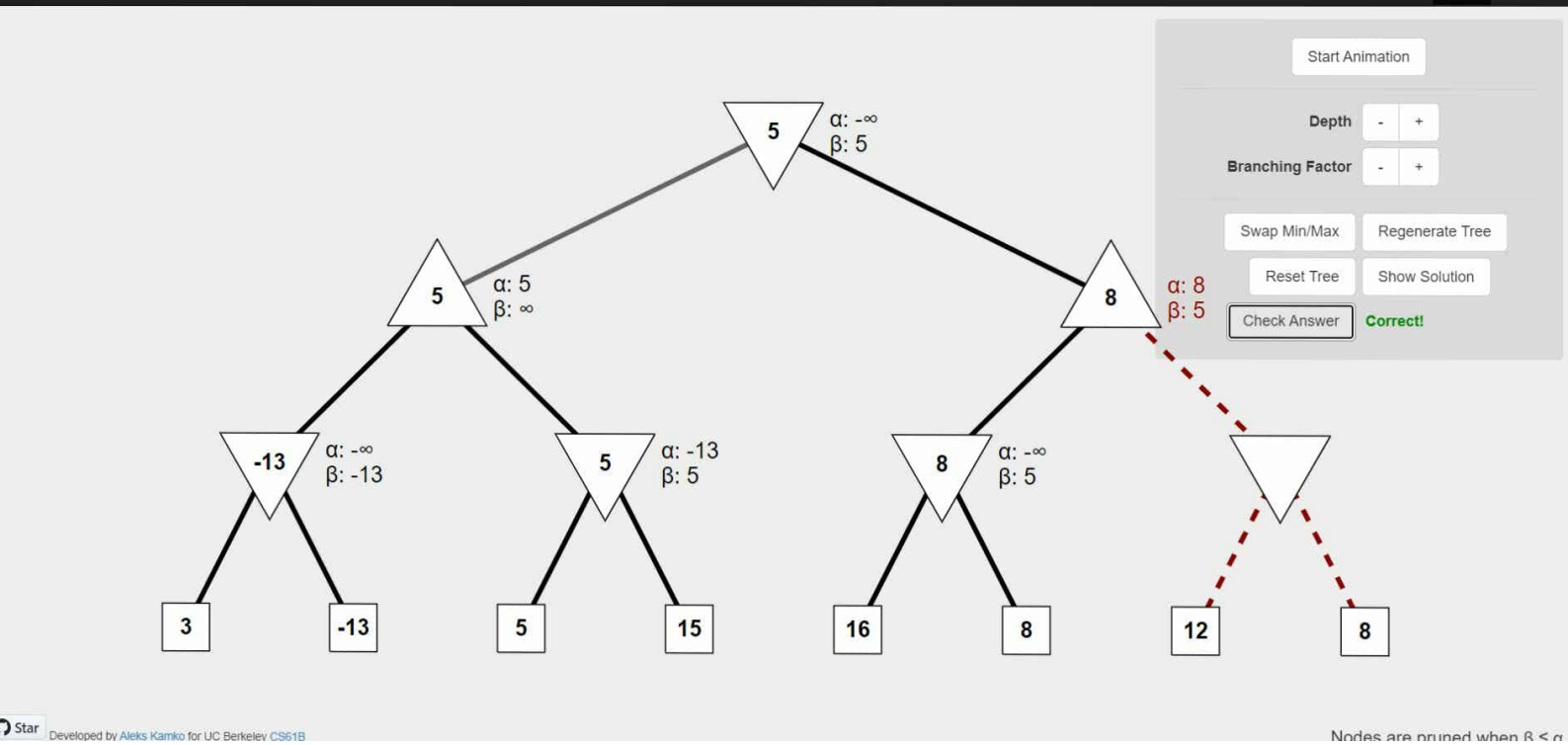
Check Answer | Correct!

Developed by Aleks Kamko for UC Berkeley CS61B

Nodes are pruned when β ≤ α.

Alpha-Beta Pruning Practice

αβ Cutoff



Star Developed by Aleks Kamko for UC Berkeley CS61B

Nodes are pruned when β ≤ α