Alpha - beta - Pruning

Module - 3

:> Name :- Tejas Balaso Jadhav.

:> Roll No. :- 24

:> Sub :- IS-LAB

:> Class :- B.E-I.T.

:> Sem :- VII

Alpha-beta-Pruning

Assignment
Module 3

Q) Mimax algorithm with alpha-beta pruning.

Ans:- 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 which are not really affecting the final decision but making algorithm slow. Hence by pruning these nodes, it makes the algorithm.

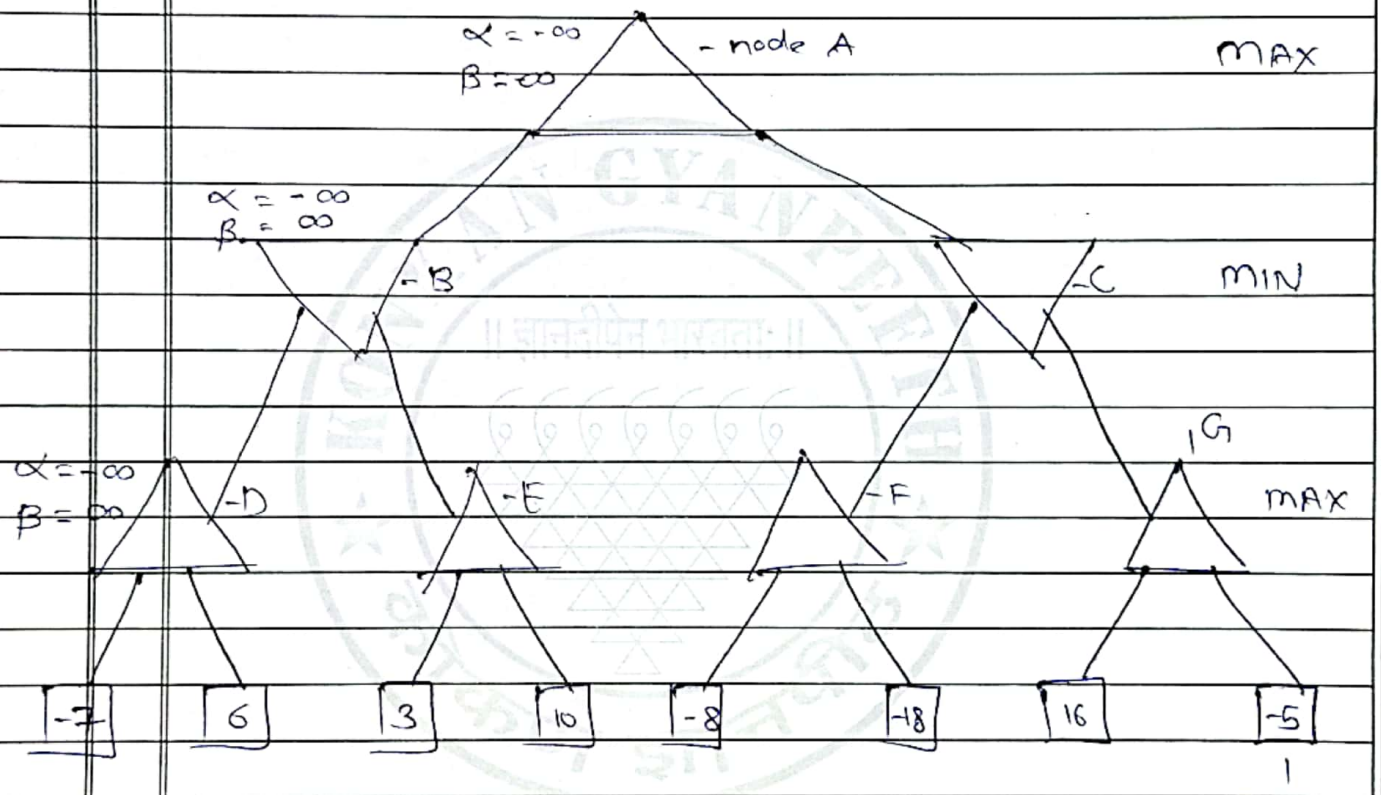
Rules & Conditions

- The Maxplayer will only update the value of alpha.
- The minplayer 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 & beta.

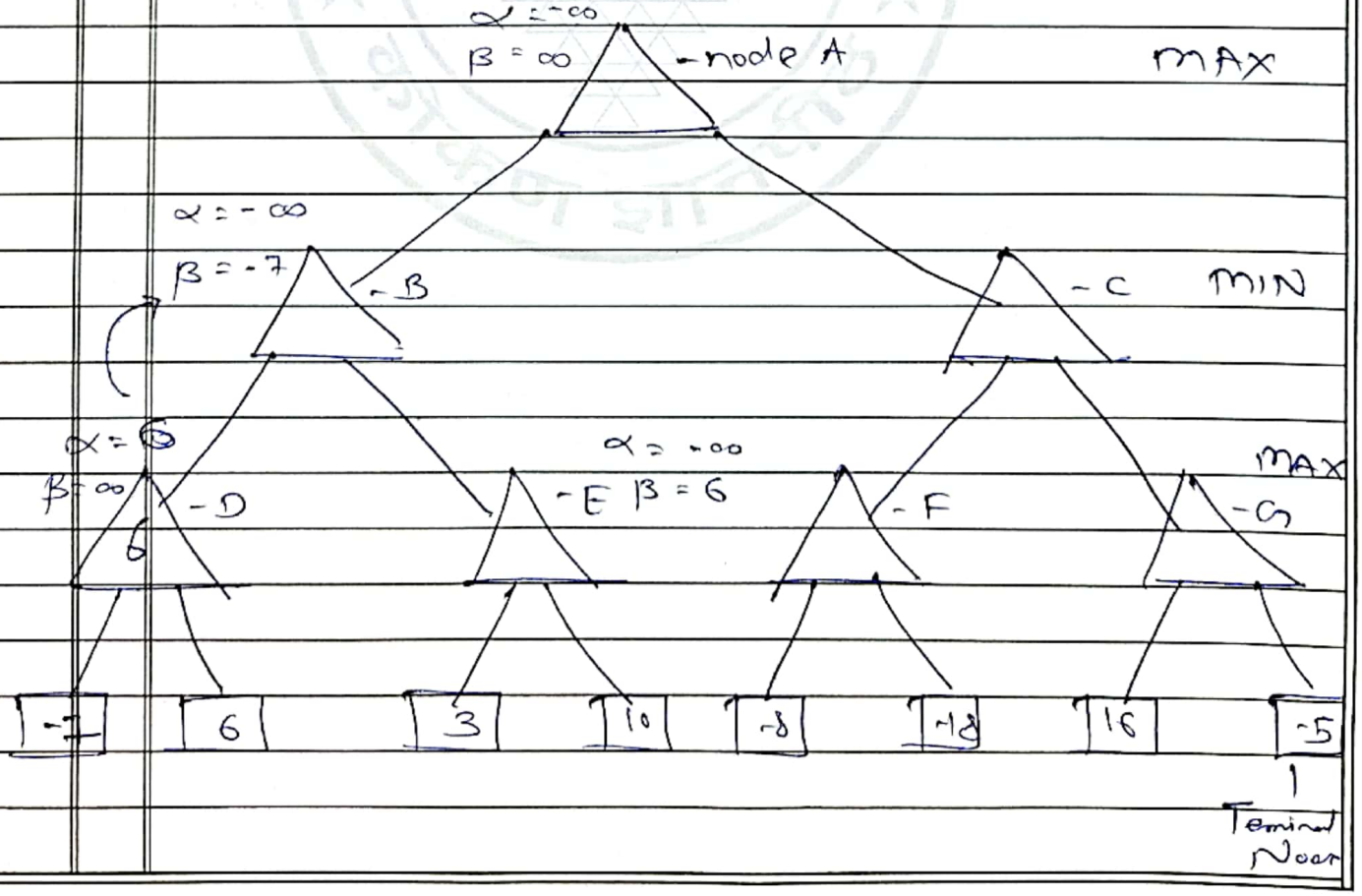
Condition to prune : $\alpha \geq \beta$ or $\beta \leq a$



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

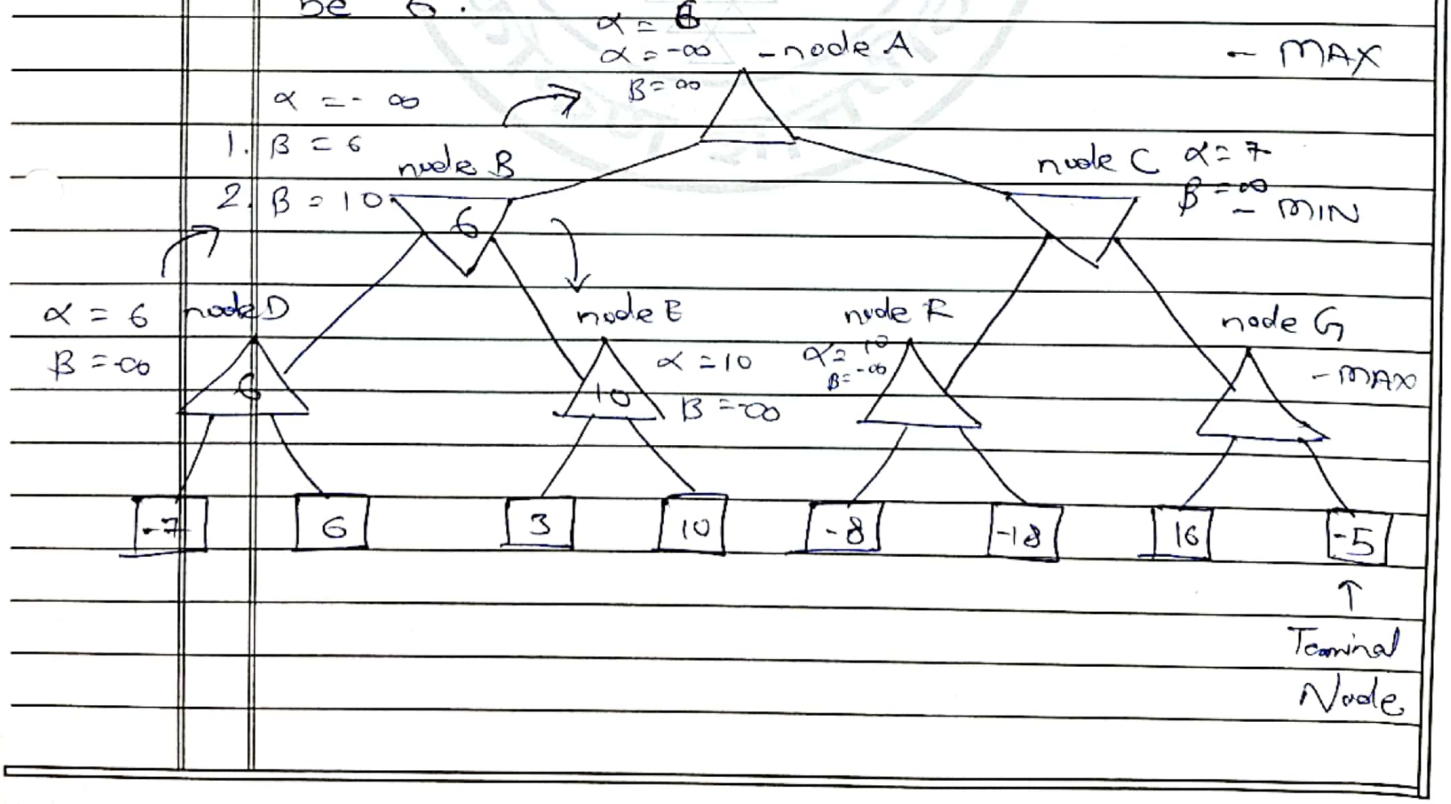
- Step 2 :- At Node D, the value of α will be calculate as its dum for max. The value of α is compared with firstly -7 & 6 & the max (-7, 6) = 6 will be the value of α at node D & node value will 6.

- Step 3 :- Now algorithm backwards to node B where the value of B will change as this is a dum of min, Now B = ∞ will compare with the available subsequent nodes value i.e.
min (∞, -7) = -7 hence at nod B
now, α = -∞, B = ~~+∞~~ -7



- Step 4 :- At the node E, max will take its turn and the value of alpha will chang the Cement value of α will be compared with 3 so max $(-\infty, 3)$ will be 3 hence at α will be compared with 10 so max (3, 10) is 10.
So the value of $\alpha = 10$ & $\beta = 6$ so at nod E value will be 10

For node B, its Min's turn so the value of B will be changed, so early of B was 6 now
B = min (6, 10) = ~~10~~ 6
So the value at node B will be 6.



Step 5 :-

At next step, algorithm again back track the tree from Node B to Node A.

The value of α will be changed the max. value will be,

α = max $(-\infty, 6)$ = ~~7~~ 6

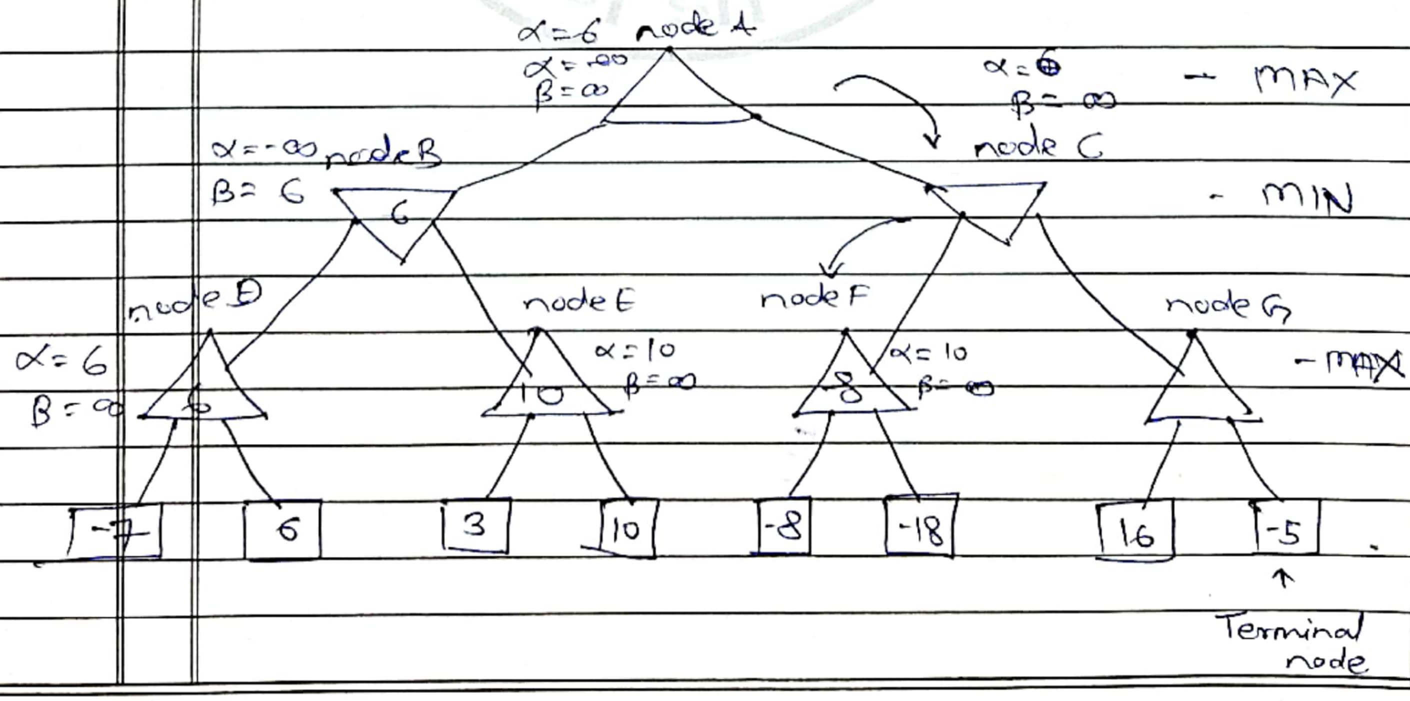
$\beta = \infty$

These two value Now pass down to the right successor which is node C. at node C $\alpha = 6$ & $\beta = \infty$ the same values will be passed to node F.

Step 6 :-

At node F again the value of α will be compared with left child & which -8 & max $(-18, -8) = -8$

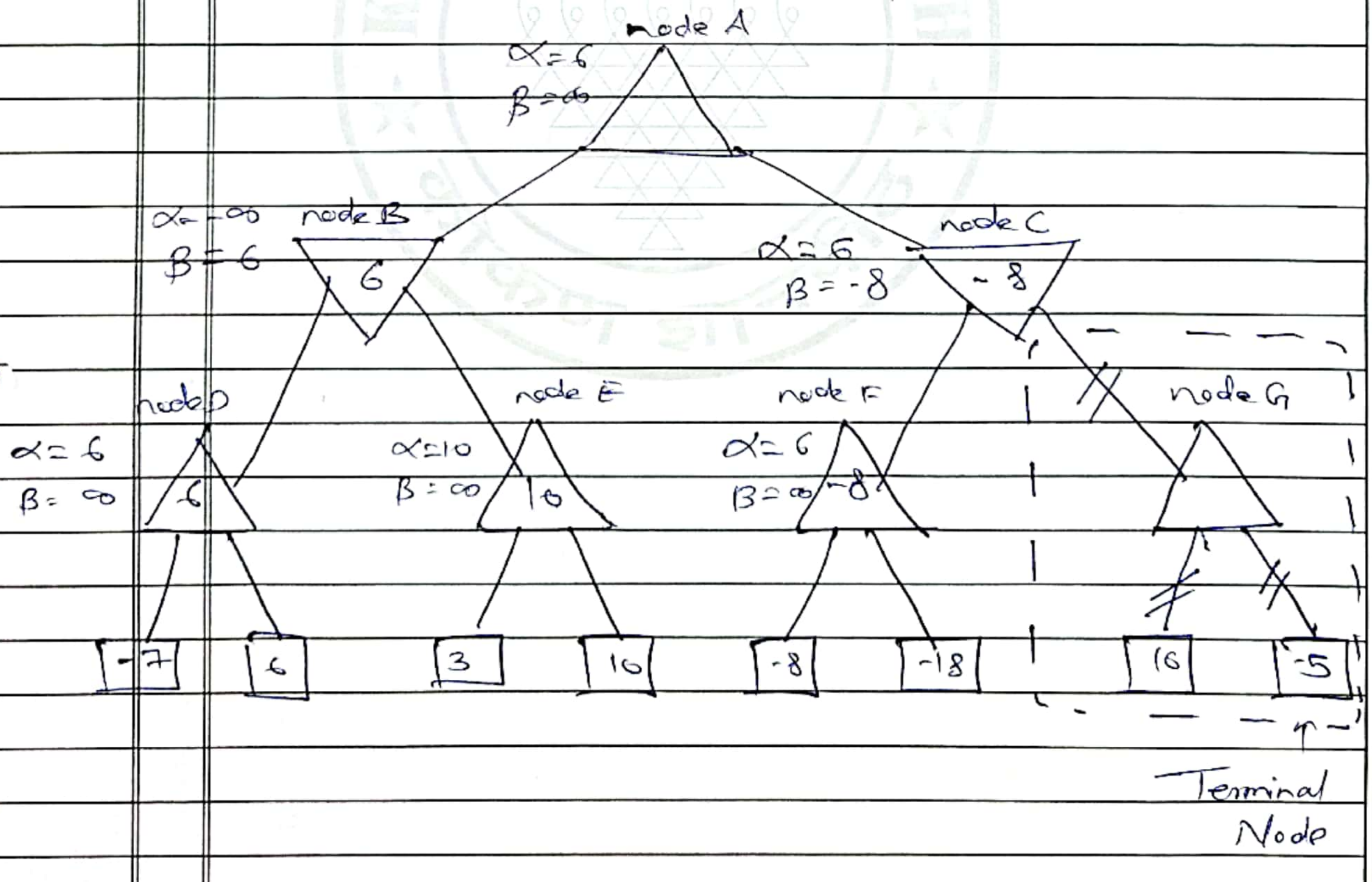
So the node value will become -8.



Step 7 :- At Node C $\alpha = 8$ & $\beta = \infty$
here the value of β will change.
it will compare with -8 So Now.

$\min(\infty, -8) = -8 = \beta$

So now we have $\alpha = 6$ & $\beta = -8$
Here the condition to prune i.e. $\alpha \geq \beta$
satisfies. So the next Right Node of
Node C will be pruned. & the
node value of C will become -8.
$\therefore$ at C $\alpha = 6$, $\beta = -8$



Step 8 :-

C Now returns the value of -8 to A here the best value of A is

$\alpha = \max(6, -8) = 6$

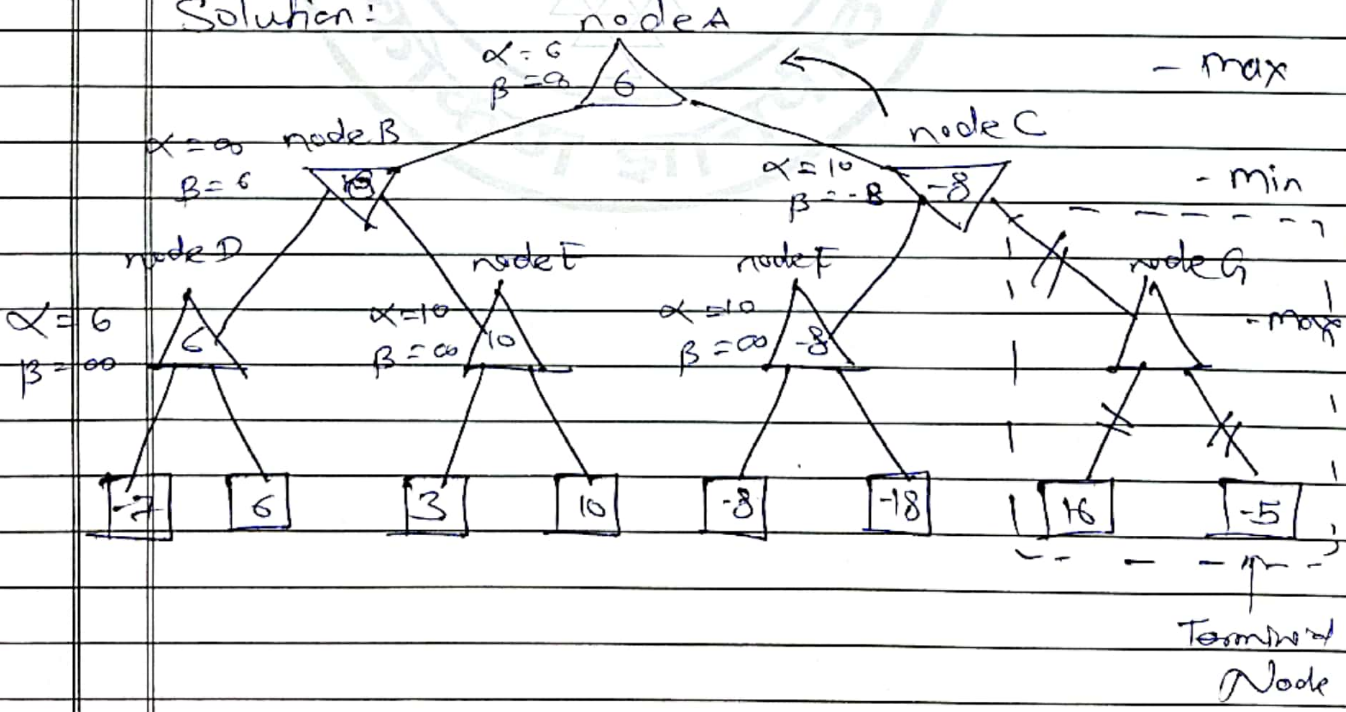
∴ So the final value of Node A will be 6

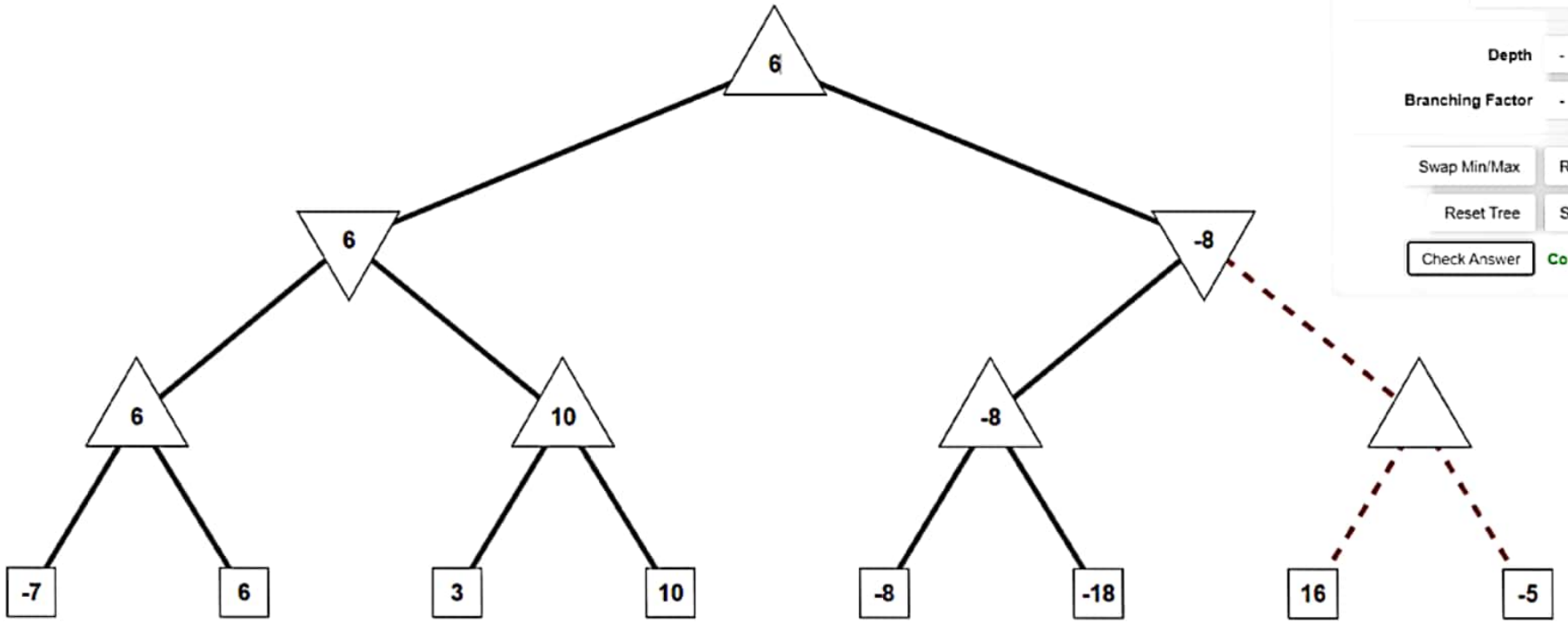
∴ $\alpha = 6$

$\beta = \infty$ at A.

Following is the final game tree which is showing the nodes which are computed. & nodes which has never computed. Hence the optimal value for the Maximum is 6 for this tree.

Solution:





Start Animation

Depth - +

Branching Factor - +

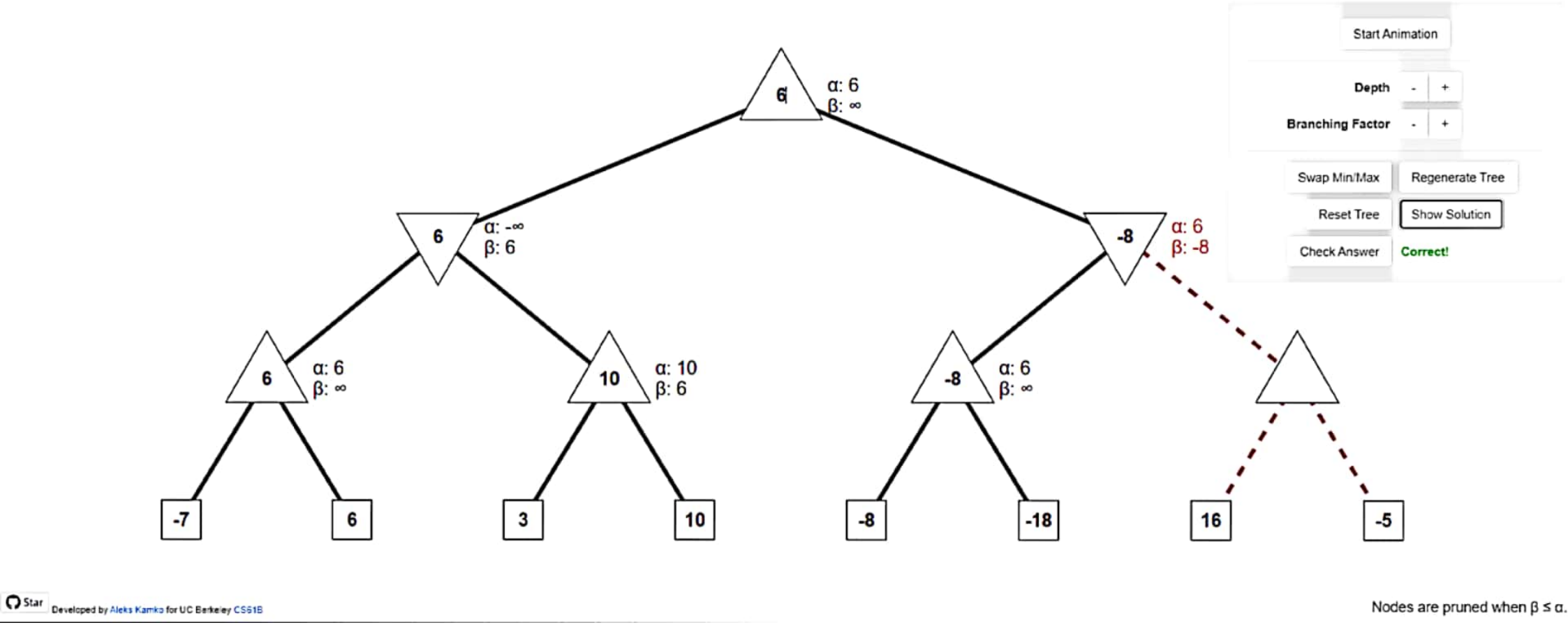
Swap Min/Max | Regenerate Tree

Reset Tree | Show Solution

Check Answer Correct!

Star Developed by Aleks Kamko for UC Berkeley CS61B

Nodes are pruned when $\beta \leq \alpha$.



Star Developed by Aleks Kamko for UC Berkeley CS61B

Nodes are pruned when $\beta \leq \alpha$.