Date
Page

Name : Gopal Arun patil

Roll No : 45

class : B.E.I.T

Sub : Is lab

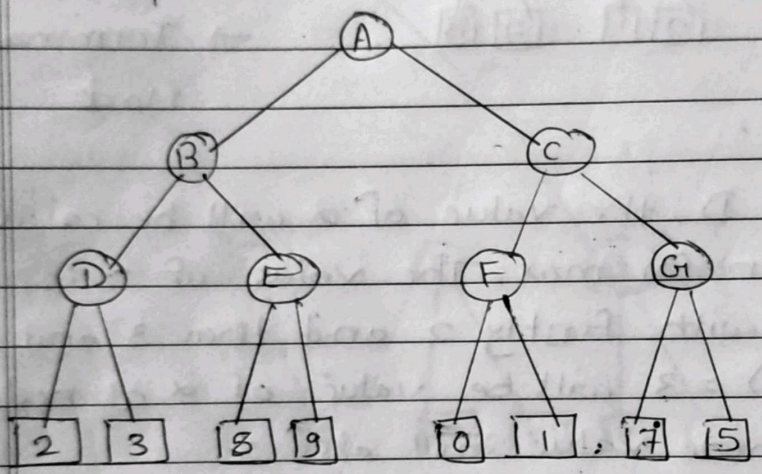
DOP	Doc	marks	sign

Date / /
Page

* Minimax Algorithm with Alpha-beta pruning

- ~~Examp~~ Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.

* Example :-



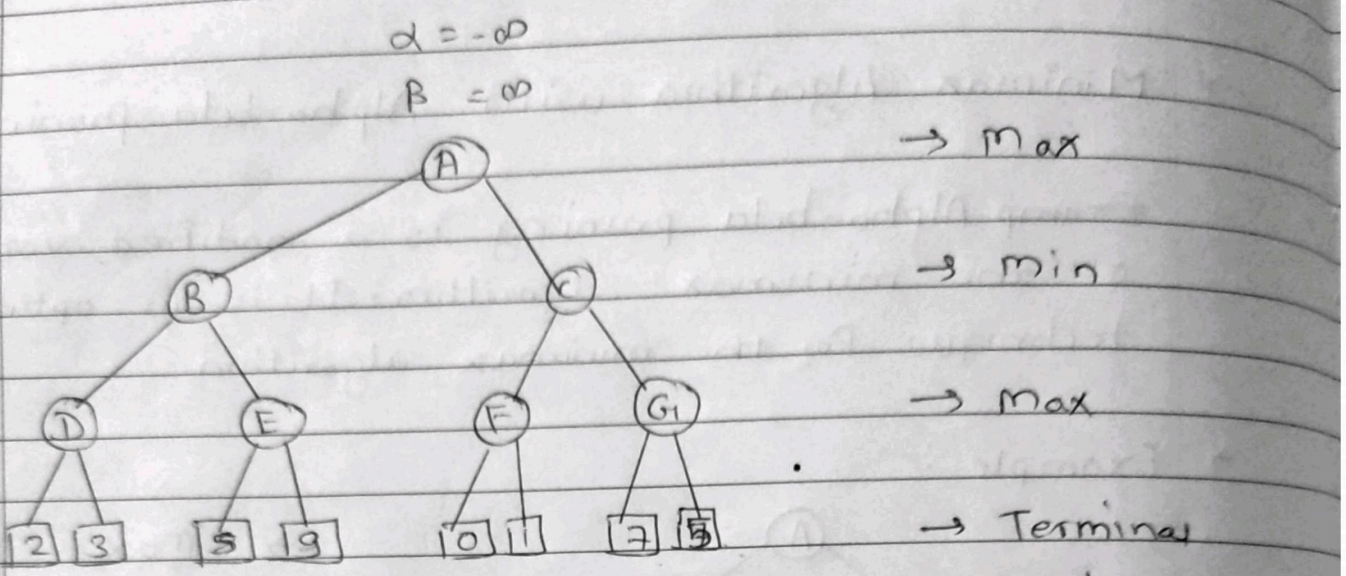
→ Max

→ Min

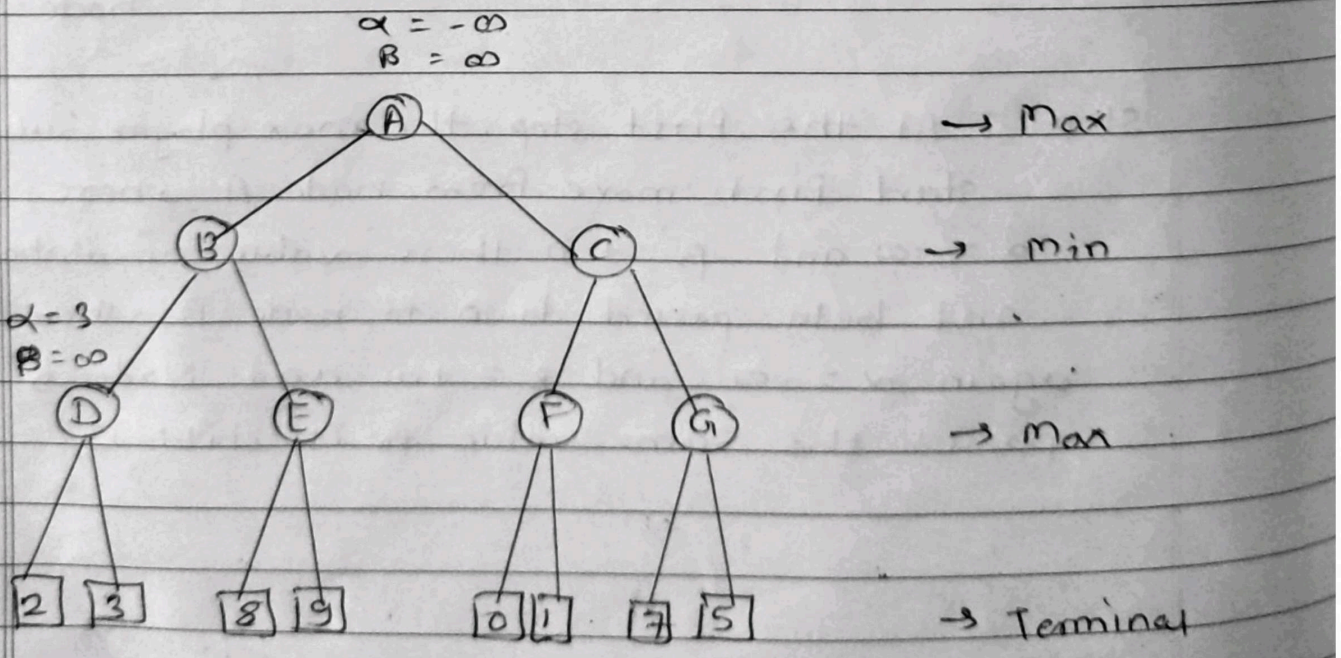
→ Max

→ Terminal node

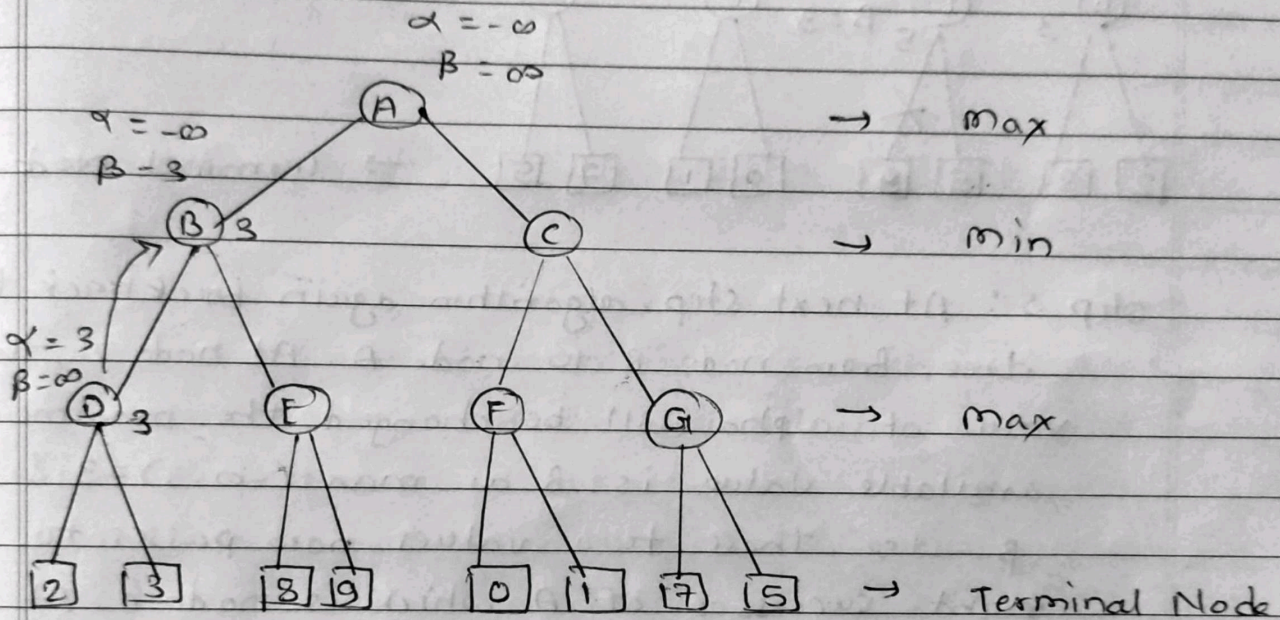
Step 1 : At the first step, the max player will start first move from node A where $\alpha = -\infty$ and $\beta = +\infty$ these value of alpha and beta passed down to node B where again $\alpha = -\infty$ and $\beta = +\infty$ and Node B passes the same value to its child D.



Step 2 : At Node D, the value of α will be calculated as its turn for max. The value of α is compared with firstly 2 and then 3 and the max (2, 3) = 3 will be value of α at node D and node value will also 3.

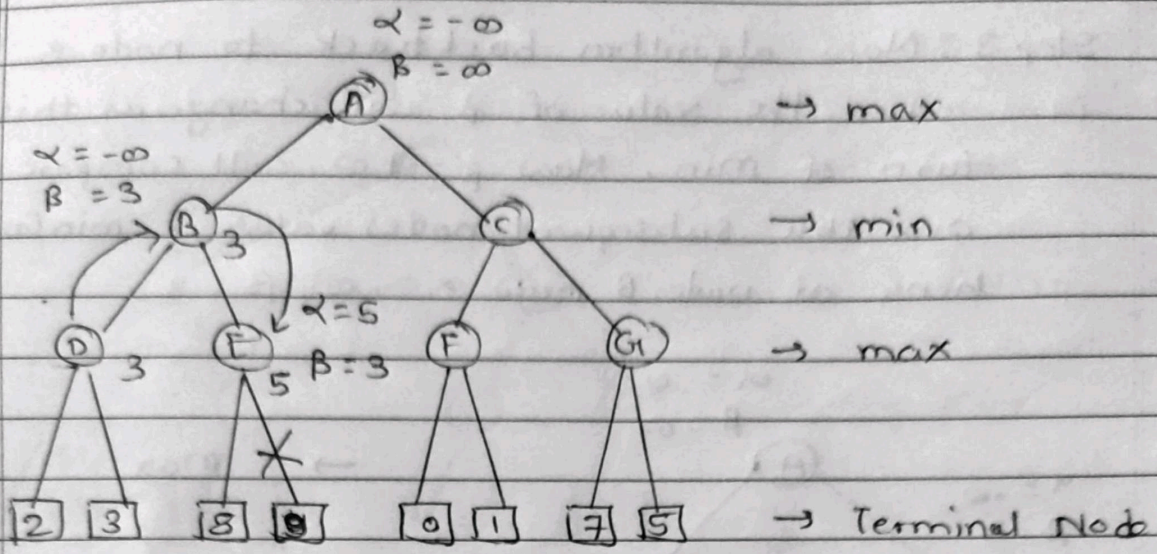


Step 3 : Now algorithm backtrack to node B, where the value of β will change as this is turn of Min, Now β = +∞, will compare with available subsequent nodes value, i.e. min(∞, 3) = 3, hence at node B now α = -∞, β = 3

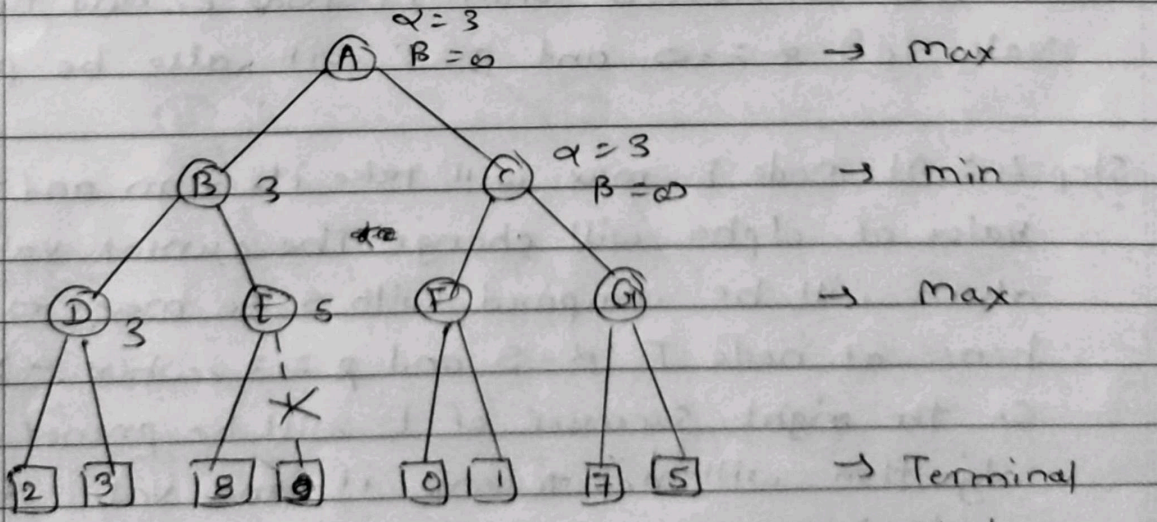


In the next step, algorithm traverse the next successor of Node B which is node E, and the value of α = -∞ and β = 3 will also be passed

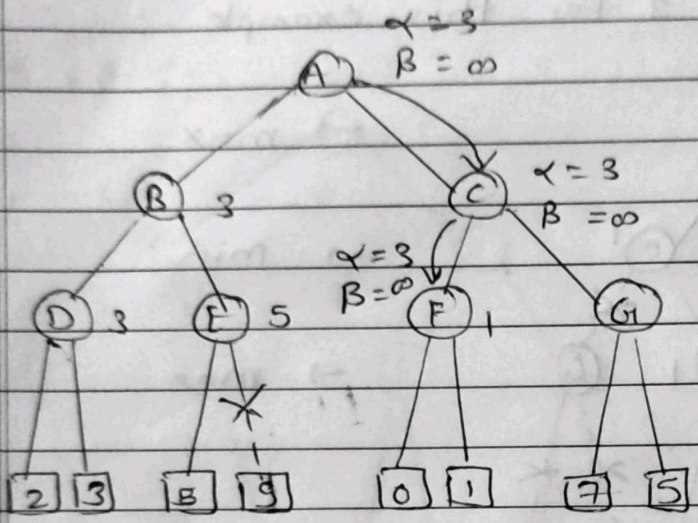
Step 4 : At node E, max will take its turn and the value of alpha will change. The current value of alpha will be compared with 5, so max(-∞, 5) = 5, hence at node E α = 5 and β = 3, where α >= β, so the right successor of E will be pruned and algorithm will not traverse it and value at node E will be 5.



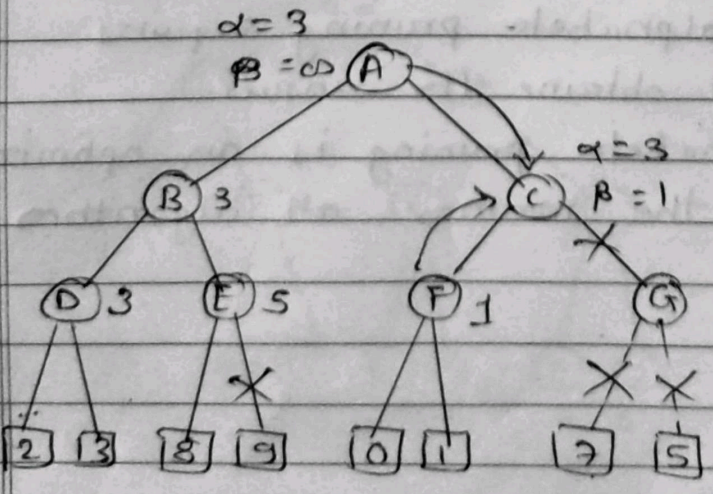
step 5: At next step, algorithm again backtrack the tree, from node B to node A. At node A, the value of alpha will be changed the maximum available value is 3 as max (-∞, 3) = 3 & β = +∞, these two values now passes to right successor of A which is node C. At node C, α = 3 & β = +∞, and the same value will be passed on to node F.



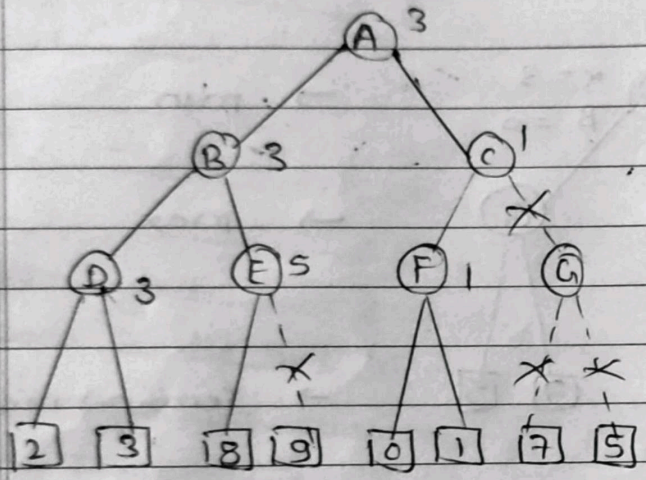
Step 6 : At node F, again the value of α will be compared with left child which is 0, max(3,0)= 3, then compared with right child which is 1, and max(3,1) = 3 still α remains 3, but the node value of F will become 1.



Step 7 : Node F returns the node value 1 to node C, at C $\alpha = 3, \beta = +\infty$, ~~there~~ the value of beta will be changed, it will compare with 1 so min(∞, 1) = 1. Now at C, $\alpha = 3$ and $\beta = 1$ and again it satisfies the condition $\alpha >= \beta$. so next child of C which is G will be pruned and algorithm will not compute the entire sub-tree G.



Step 8 :C, Now return value of 1 to A here the best value for A is max (3,1) = 3. Following is the final game tree which is showing nodes which are computed and nodes which has never computed. Hence the optimal value for the maximizer is 3 for this example



Conclusion : Minmax algorithm using alphabeta pruning is an optimization technique for the minimax algorithm

① In this example min-max algorithm requires 7 steps to obtain the final answer where as minimax using alphabeta pruning requires 6 steps to get obtaine the answer.

② Therefore alphabeta pruning is an optimization technique for the minmax algorithm.