Q1-1: Which is true about the two approaches to compute the value on the initial node of a game tree?

1. The DFS implementation of minimax search has better time complexity than the bottom up approach
2. The DFS implementation of minimax search has better space complexity than the bottom up approach
3. Both 1 and 2
4. None of the above

Q1-1: Which is true about the two approaches to compute the value on the initial node of a game tree?

1. The DFS implementation of minimax search has better time complexity than the bottom up approach
2. The DFS implementation of minimax search has better space complexity than the bottom up approach
3. Both 1 and 2
4. None of the above

Q1-2: Which is true about the DFS implementation of minimax search? Suppose it evaluates the children from left to right.

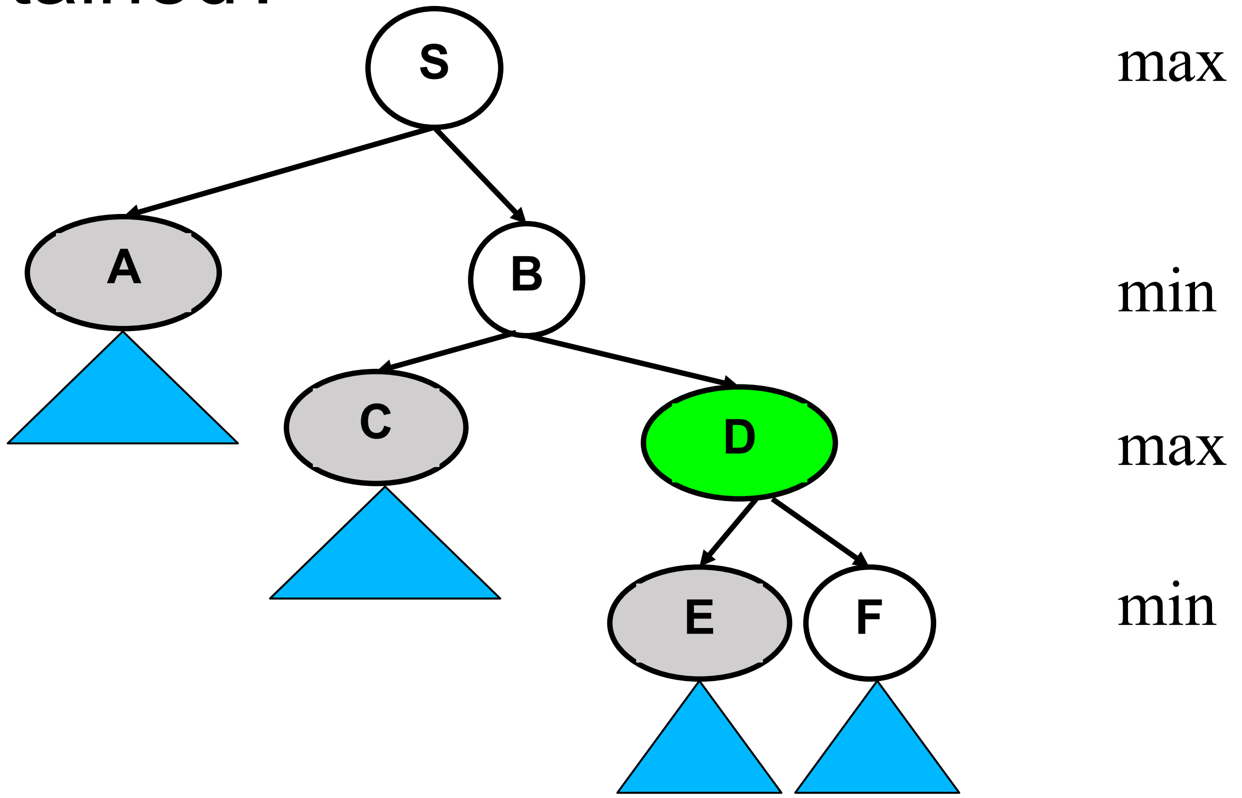
1. It will visit the leaves in the subtree of a left child before visiting a right child
2. It will finish computing the value of a left child before visiting a right child
3. Both 1 and 2
4. None of the above

Q1-2: Which is true about the DFS implementation of minimax search? Suppose it evaluates the children from left to right.

1. It will visit the leaves in the subtree of a left child before visiting a right child
2. It will finish computing the value of a left child before visiting a right child
3. Both 1 and 2
4. None of the above

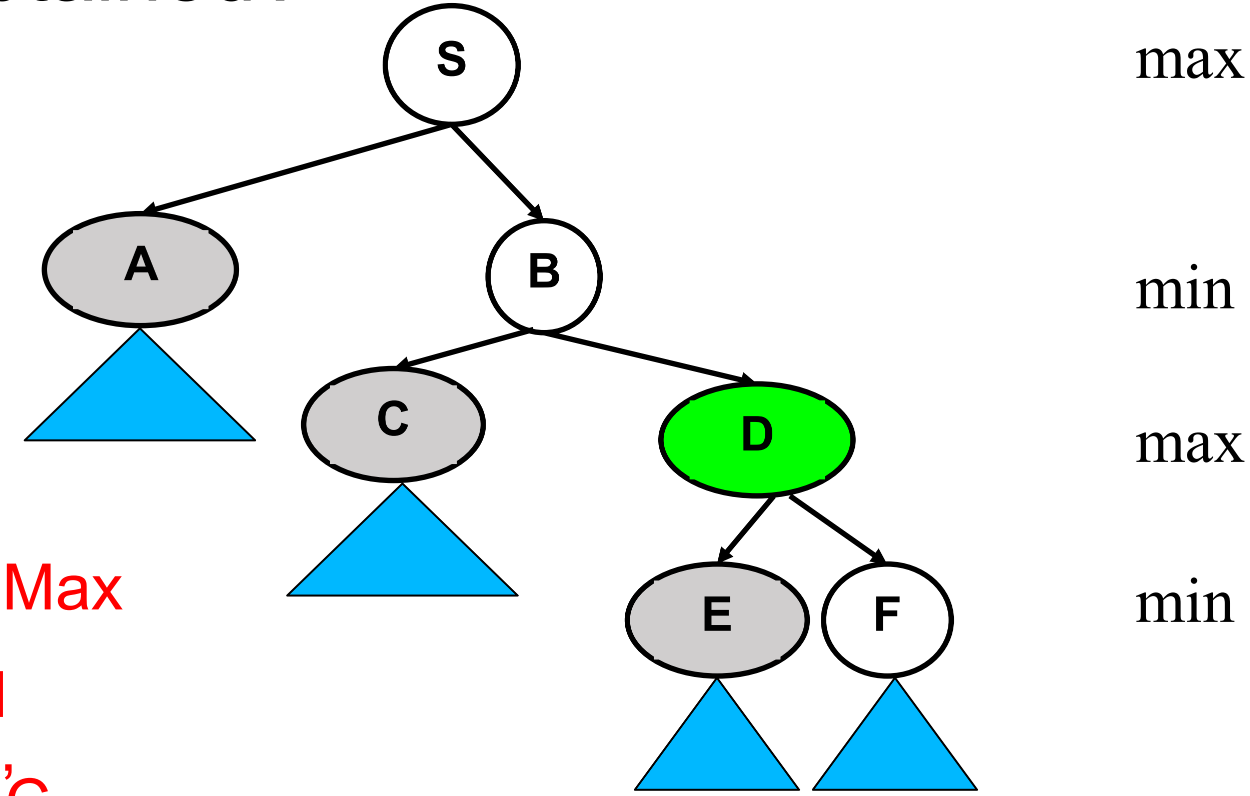
Q1-3: Suppose the minimax search evaluates the children from left to right. It has computed the value of E and returned to D but hasn't visited F. Up to now, the best value Max can make sure is X (no matter what subtree of F looks like, Max has a way to get a score >=X). Where can X be obtained?

1. X can be the value of A or E
2. X can be the value of C
3. X can be the value of B or D
4. Both 1 and 2



Q1-3: Suppose the minimax search evaluates the children from left to right. It has computed the value of E and returned to D but hasn't visited F. Up to now, the best value Max can make sure is X (no matter what subtree of F looks like, Max has a way to get a score >=X). Where can X be obtained?

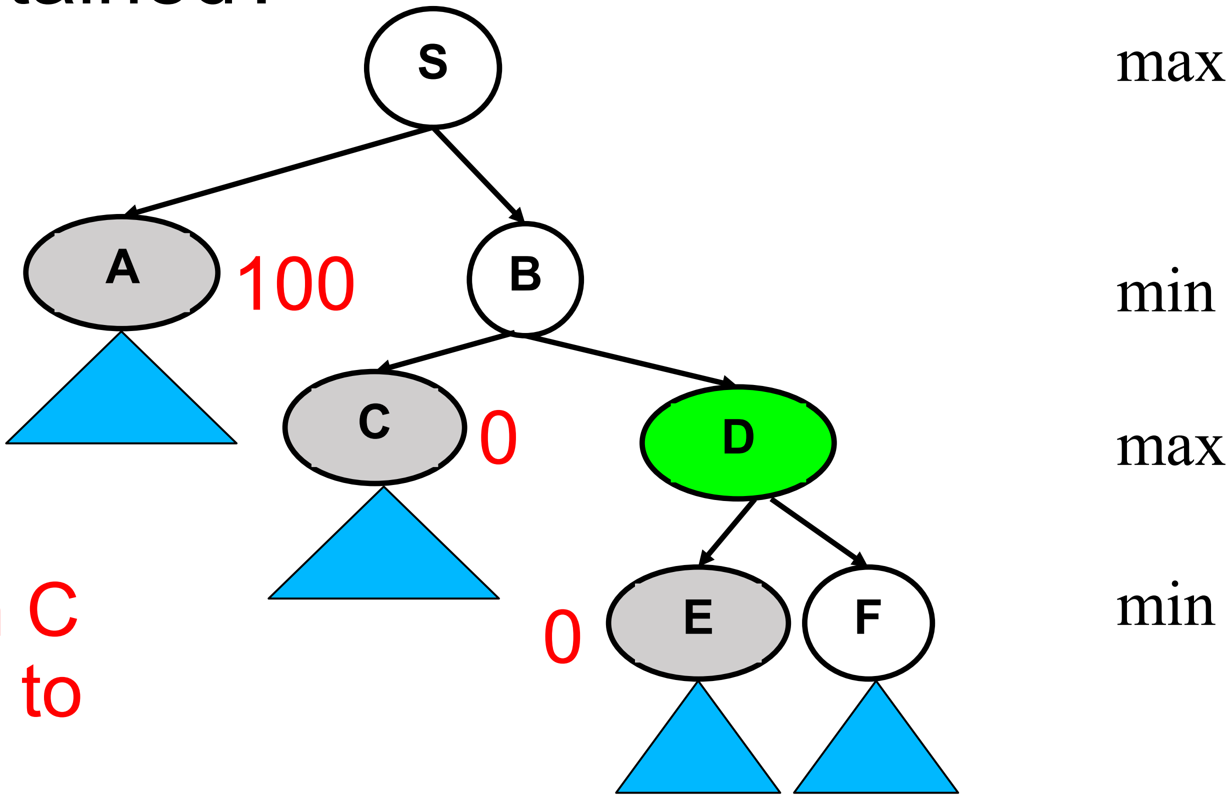
1. X can be the value of A or E
2. X can be the value of C
3. X can be the value of B or D
4. Both 1 and 2 ⟵



If A's value is larger than C and E, then Max can choose to go to A. If the values are C>E>A, Max can choose to go to B and guarantees at least E's value. If E>C>A, then Max will go to B and min will go to C, so X is obtained on C. The value of B or D has not been computed yet.

Q1-3: Suppose the minimax search evaluates the children from left to right. It has computed the value of E and returned to D but hasn't visited F. Up to now, the best value Max can make sure is X (no matter what subtree of F looks like, Max has a way to get a score >=X). Where can X be obtained?

1. X can be the value of A or E
2. X can be the value of C
3. X can be the value of B or D
4. Both 1 and 2 ←

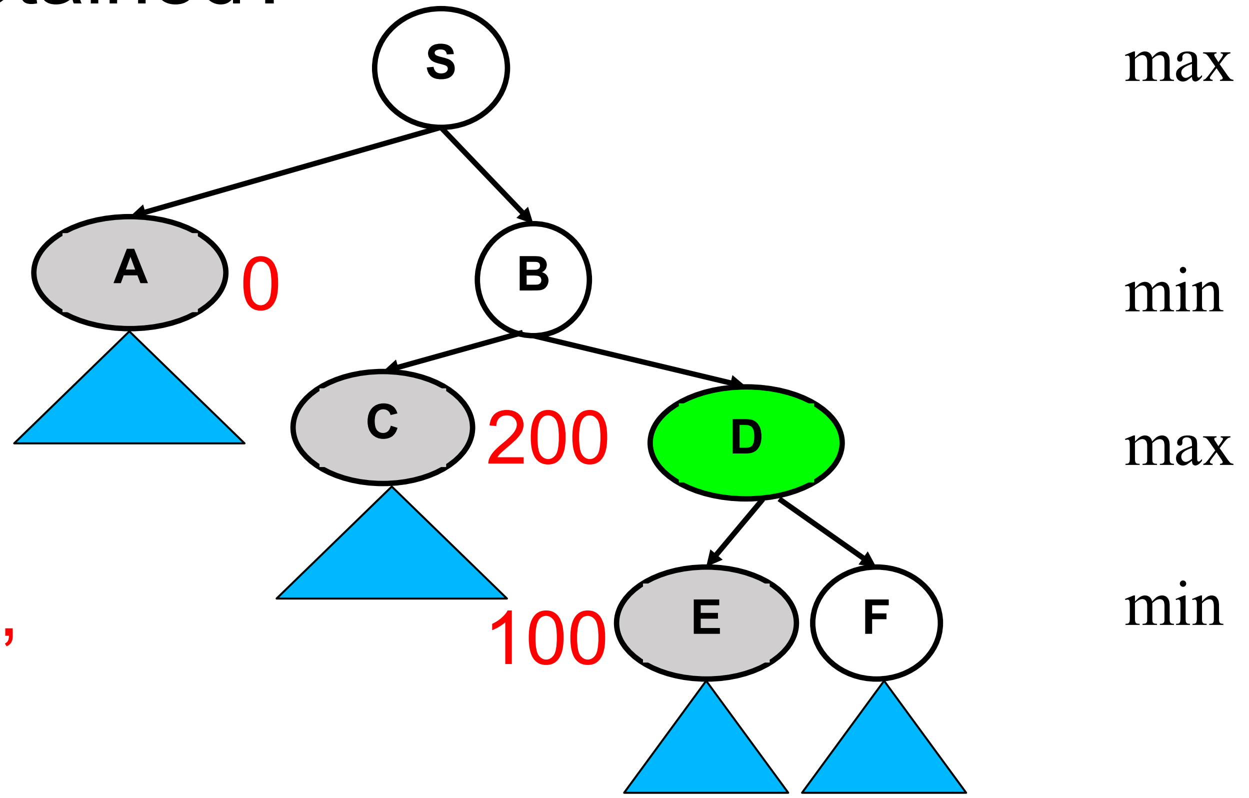


Example: If A's value is larger than C and E, then Max can choose to go to A.

Q1-3: Suppose the minimax search evaluates the children from left to right. It has computed the value of E and returned to D but hasn't visited F. Up to now, the best value Max can make sure is X (no matter what subtree of F looks like, Max has a way to get a score >=X). Where can X be obtained?

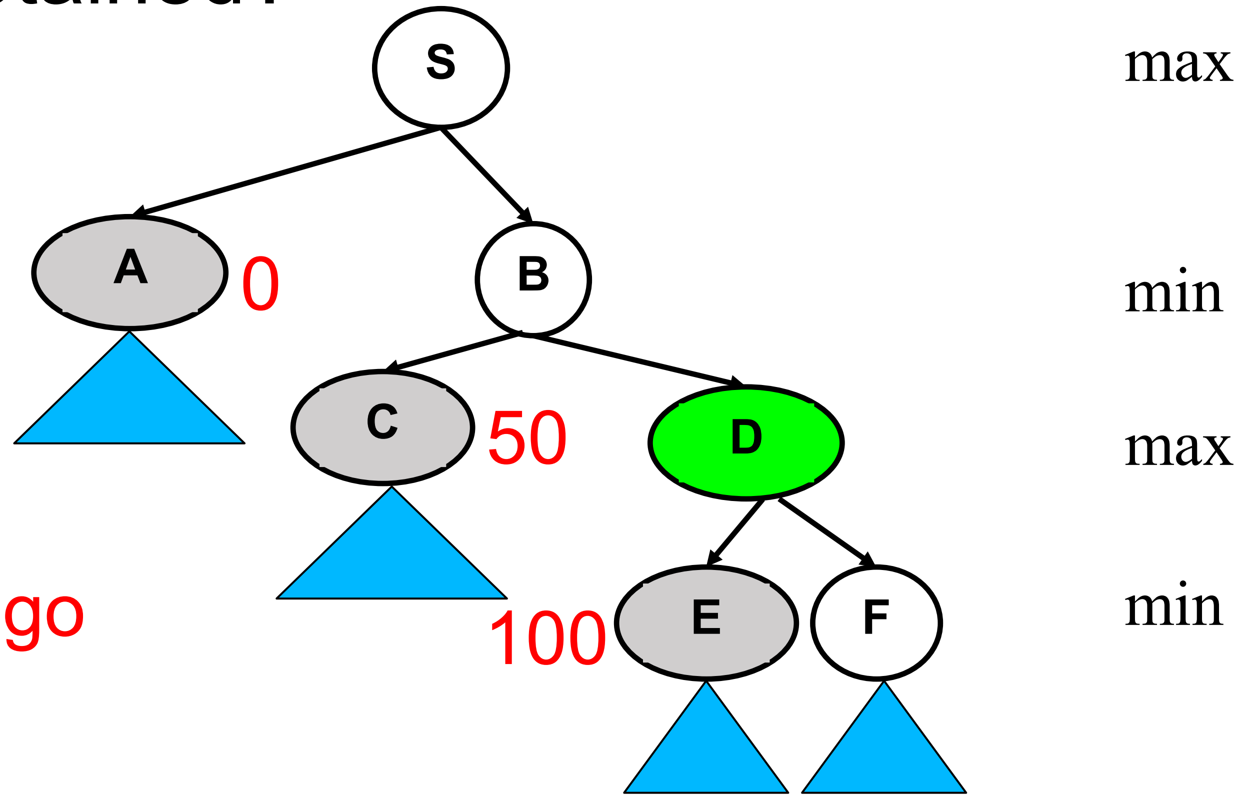
1. X can be the value of A or E
2. X can be the value of C
3. X can be the value of B or D
4. Both 1 and 2 ⟵

Example: If the values are C>E>A, Max can choose to go to B and guarantees at least E's value.



Q1-3: Suppose the minimax search evaluates the children from left to right. It has computed the value of E and returned to D but hasn't visited F. Up to now, the best value Max can make sure is X (no matter what subtree of F looks like, Max has a way to get a score >=X). Where can X be obtained?

1. X can be the value of A or E
2. X can be the value of C
3. X can be the value of B or D
4. Both 1 and 2 ←



Example: If E>C>A, then Max will go to B and min will go to C, so X is obtained on C.

Q2-1: Under which of the circumstance can the alpha on a max node or the beta value on a min node be determined (i.e., not infinity)?

A. all leaves under that node must have been evaluated
B. all subtree under that node must have been evaluated
C. at least a leave under that node have been evaluated
D. at least a subtree under that node have been evaluated

Q2-1: Under which of the circumstance can the alpha on a max node or the beta value on a min node be determined (i.e., not infinity)?

A. all leaves under that node must have been evaluated
B. all subtree under that node must have been evaluated
C. at least a leave under that node have been evaluated
D. at least a subtree under that node have been evaluated

Q2-2: In which of the following situations, we can prune some subtree? (multiple correct answers)

A. On a max node, its alpha is larger than its parent's beta
B. On a min node, its beta goes below its parent's alpha
C. On a max node, its alpha is larger than its parent's alpha
D. On a min node, its beta goes below its parent's beta

Q2-2: In which of the following situations, we can prune some subtree? (multiple correct answers)

A. On a max node, its alpha is larger than its parent's beta ⬅
B. On a min node, its beta goes below its parent's alpha ⬅
C. On a max node, its alpha is larger than its parent's alpha
D. On a min node, its beta goes below its parent's beta

Q2-3: When on a node v, which of the following is correct regarding the alpha value on that node?

A. Alpha is the maximum value over all the leaves we've seen so far
B. Alpha is the maximum value over all the evaluated children of the nodes from root to v (regardless of max nodes or min nodes)
C. Alpha is the maximum value over all the evaluated children of the max nodes from root to v
D. Alpha is the maximum value over all the evaluated children of the min nodes from root to v

Q2-3: When on a node v, which of the following is correct regarding the alpha value on that node?

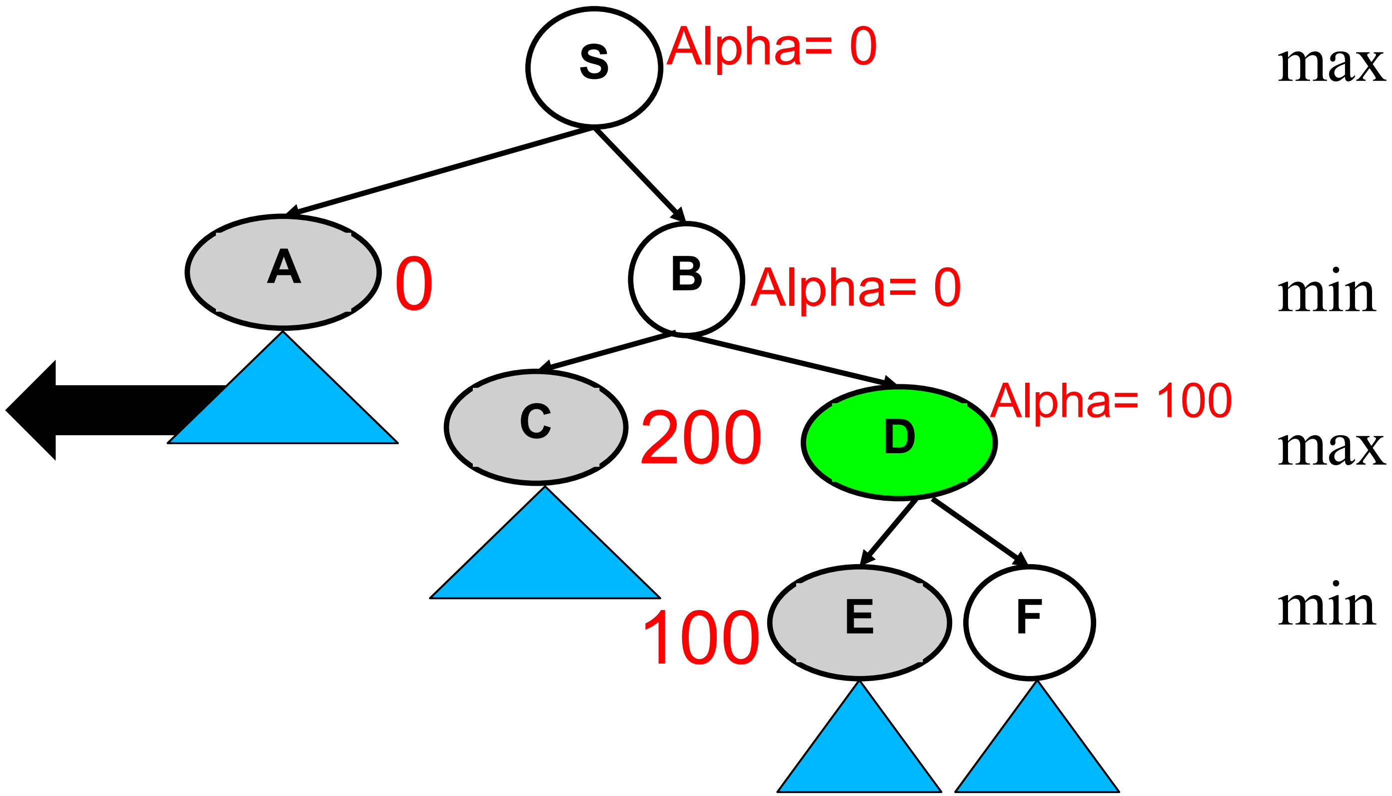
A. Alpha is the maximum value over all the leaves we've seen so far
B. Alpha is the maximum value over all the evaluated children of the nodes from root to v (regardless of max nodes or min nodes)
C. Alpha is the maximum value over all the evaluated children of the max nodes from root to v
D. Alpha is the maximum value over all the evaluated children of the min nodes from root to v

Alpha is inherited from the parent, and only get updated on max nodes using their children's values. The updates can only increase the value.

Q2-3: When on a node v, which of the following is correct regarding the alpha value on that node?

A. Alpha is the maximum value over all the leaves we've seen so far
B. Alpha is the maximum value over all the evaluated children of the nodes from root to v (regardless of max nodes or min nodes)
C. Alpha is the maximum value over all the evaluated children of the max nodes from root to v
D. Alpha is the maximum value over all the evaluated children of the min nodes from root to v

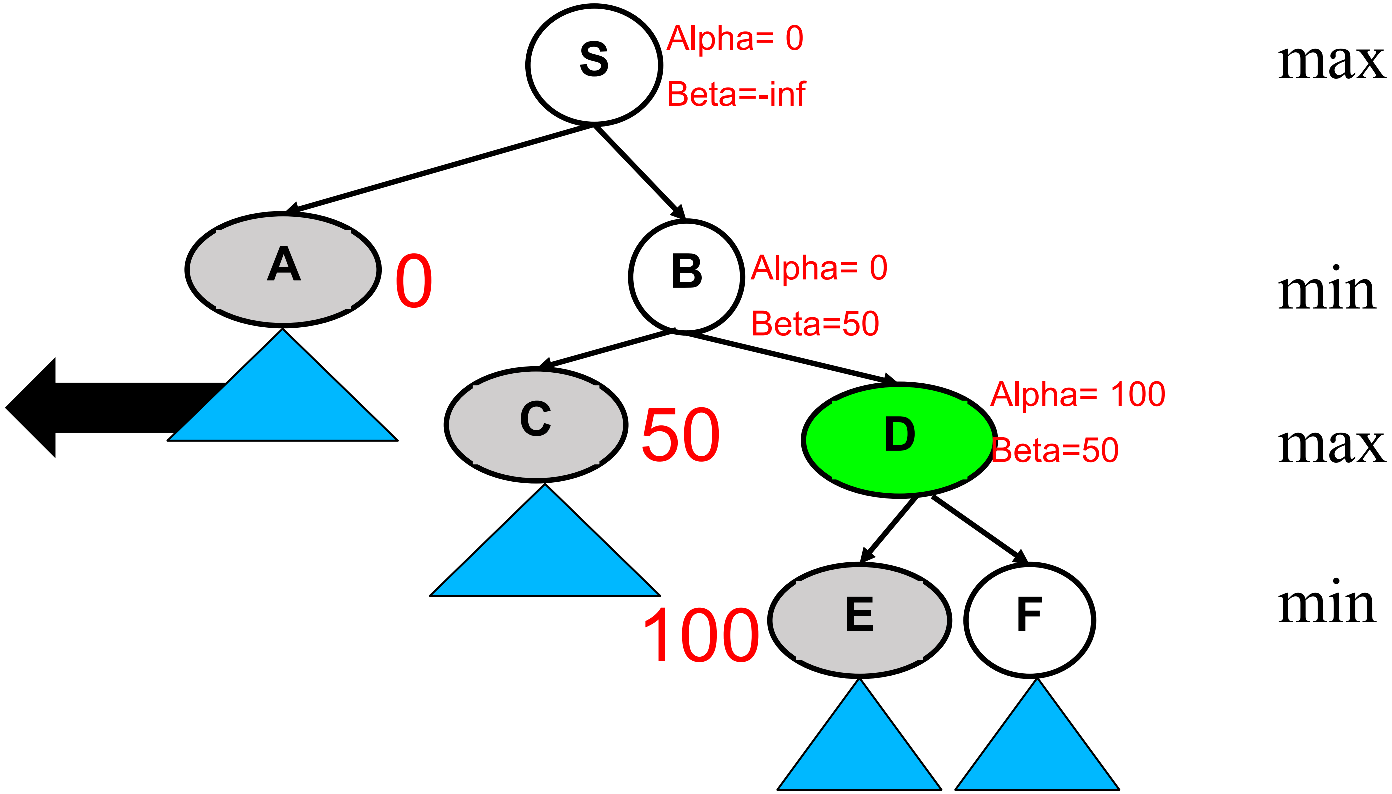
This example shows why B and D are wrong: consider C. It also shows why A is wrong: consider when C is a leaf.



Q2-3: When on a node v, which of the following is correct regarding the alpha value on that node?

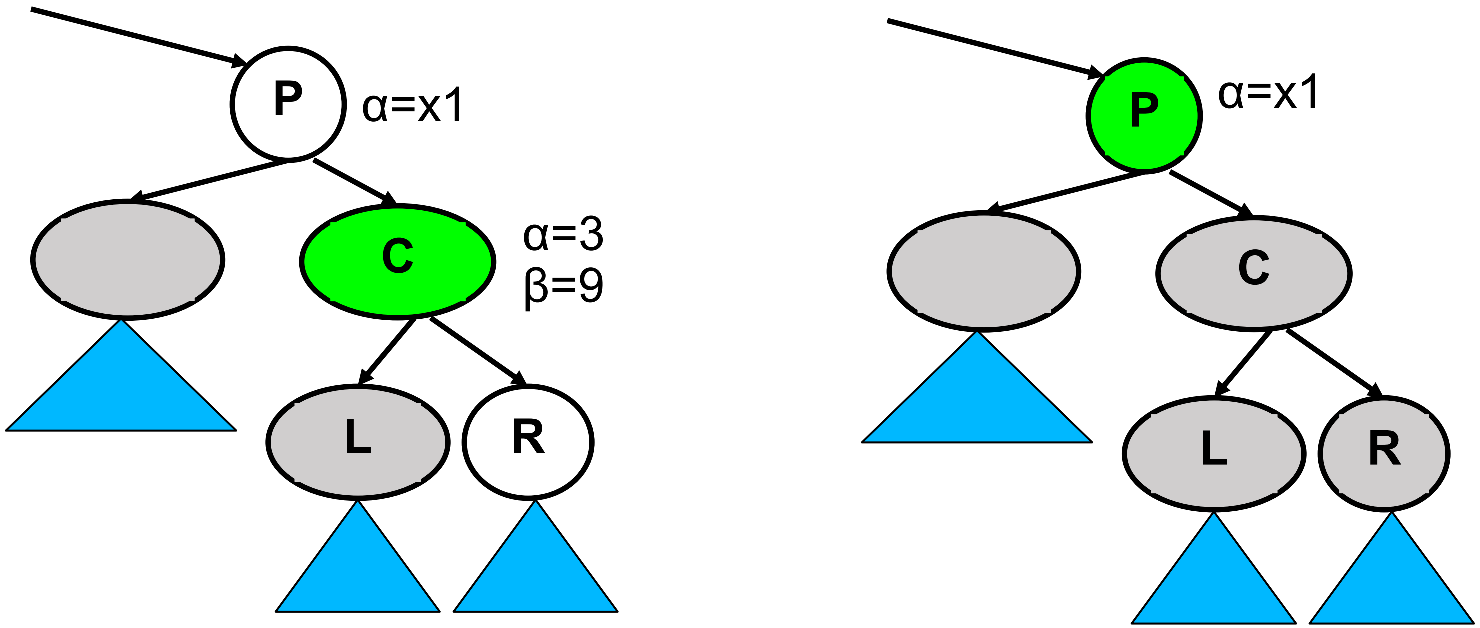
A. Alpha is the maximum value over all the leaves we've seen so far
B. Alpha is the maximum value over all the evaluated children of the nodes from root to v (regardless of max nodes or min nodes)
C. Alpha is the maximum value over all the evaluated children of the max nodes from root to v
D. Alpha is the maximum value over all the evaluated children of the min nodes from root to v

Consider another example. At this point, alpha is still the max value on the max nodes A and E. But alpha is not the best value Max can make sure, since at this point alpha>beta on D so Min won't choose to go to D.



Q3-1: We have beta=9, alpha=3 on the current node C after checking L but not R. Suppose after checking R and returning to the parent node P, the alpha on P is not updated. Which value of the node R guarantees that this happens?

A. 2
B. 4
C. 6
D. 8

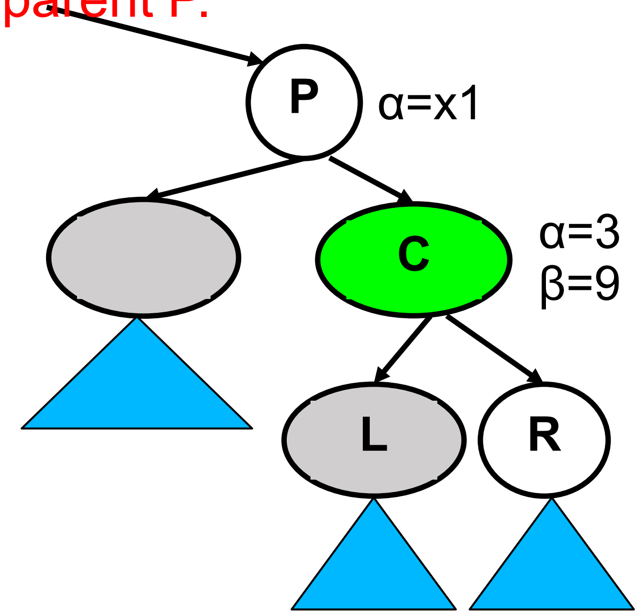


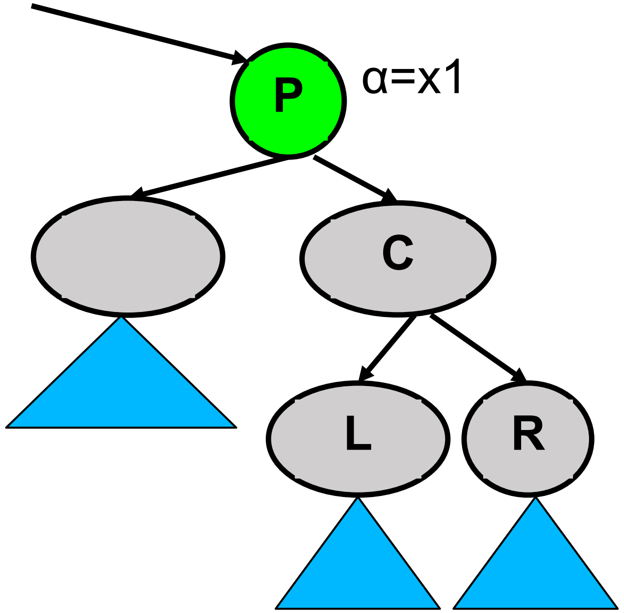
Q3-1: We have beta=9, alpha=3 on the current node C after checking L but not R. Suppose after checking R and returning to the parent node P, the alpha on P is not updated. Which value of the node R guarantees that this happens?

A. 2 ⬅
B. 4
C. 6
D. 8

Think about the execution of alpha-beta pruning.

1. If the current node is a max node where alpha is updated. Then P is a min node and only updates its beta value.
2. If the current node is a min node where beta is updated. Then x1 must be 3. Also, beta on C is updated to 2, and we return 3 to the parent P.





Q3-2: We have enough computation resource to evaluate a tree with depth m without pruning. In the worst case, what is the depth of the tree we can evaluate with alpha-beta pruning?

A. 2m
B. m
C. m^2
D. ln(m)

Q3-2: We have enough computation resource to evaluate a tree with depth m without pruning. In the worst case, what is the depth of the tree we can evaluate with alpha-beta pruning?

A. 2m
B. m
C. m^2
D. ln(m)

Q3-3: We have enough computation resource to evaluate a tree with depth m without pruning. In the best case, what is the depth of the tree we can evaluate with alpha-beta pruning?

A. 2m
B. m
C. m^2
D. ln(m)

Q3-3: We have enough computation resource to evaluate a tree with depth m without pruning. In the best case, what is the depth of the tree we can evaluate with alpha-beta pruning?

A. 2m
B. m
C. m^2
D. ln(m)