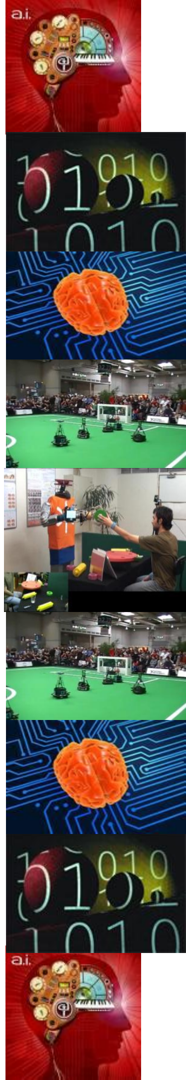
Game Trees and Game Playing

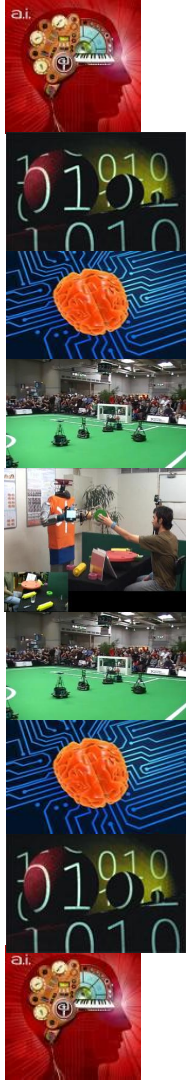
Game Playing

- Many board games have a high level of complexity.
- The problem can be modelled as one of trying to find the best choices of moves from a tree.
- The tree possibilities correspond to the board positions from ind moves

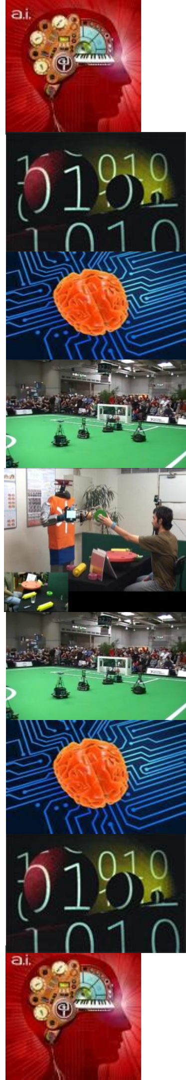


Game Tree

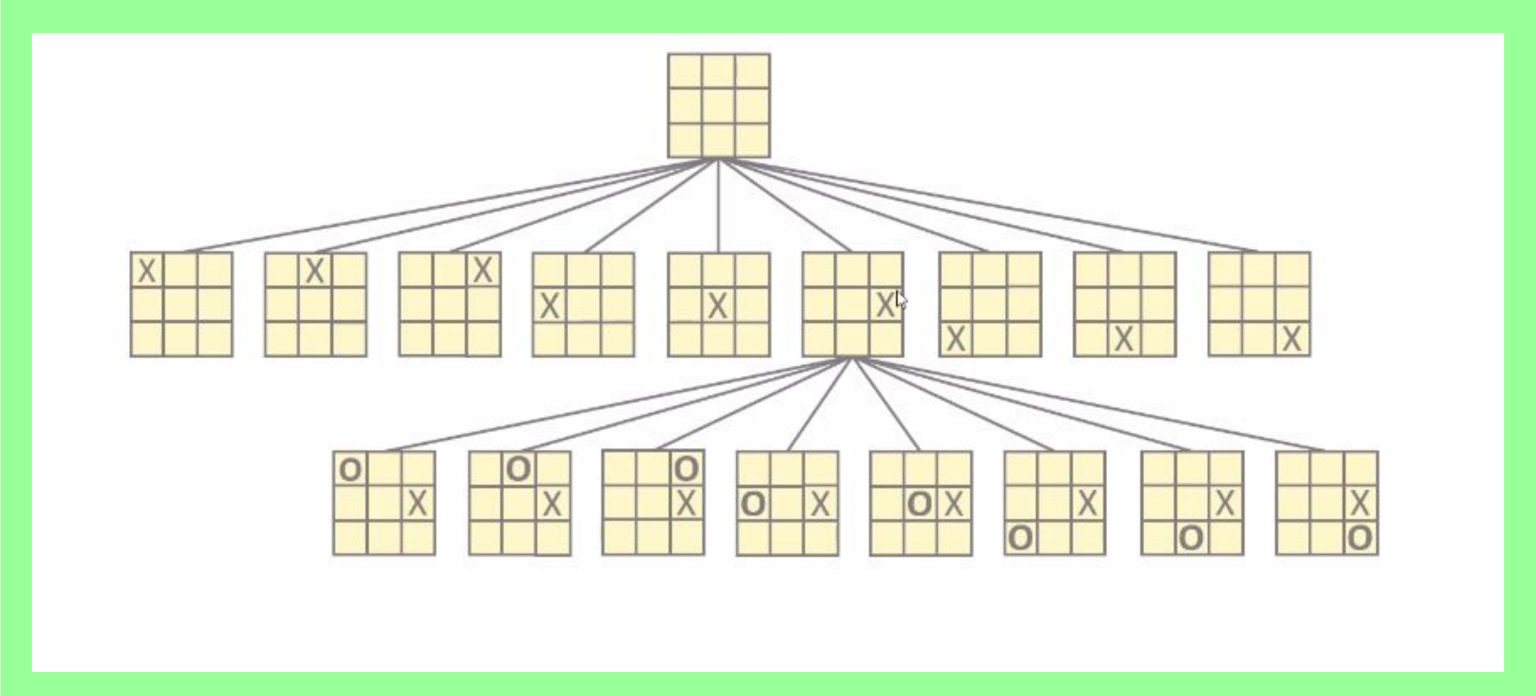
- Is a directed graph G(V,E) that represents every possible outcome.
- V- node position in the game
- E - edges moves
- Not always possible to represent all games

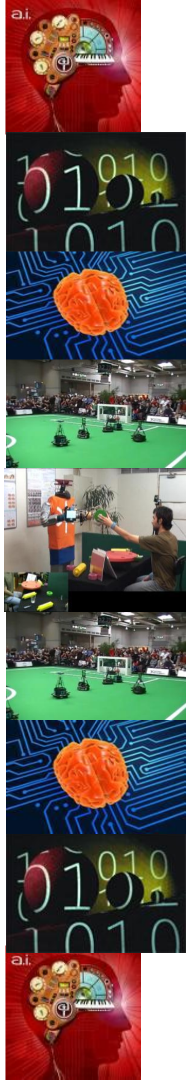


- Ideally we materialize the whole tree and select the ideal move.
- If not then a partial tree is generated.
- We need an effective search algorithm.



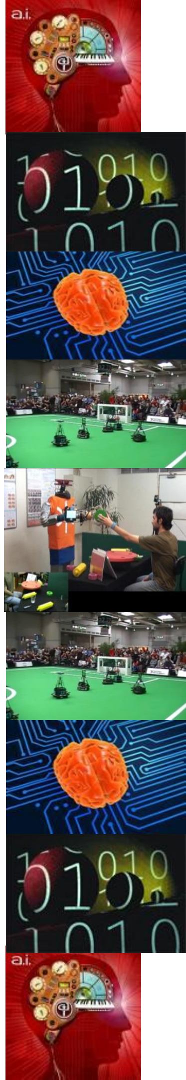
Game Tree -Tic tac toe





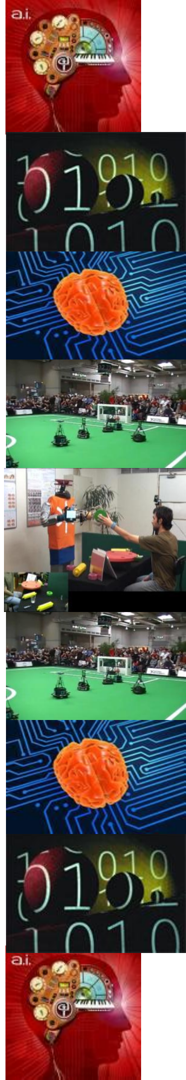
Minimax Algorithm

- An important search algorithm is the minimax algorithm.
- It is a **depth first search** algorithm.
- We evaluate the complete tree using the Minimax Algorithm to select the best moves.



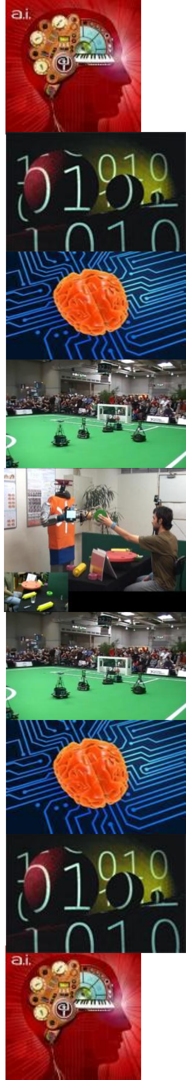
Minimax Algorithm

- The Minimax algorithm works by generation a complete tree (if possible).
- There are 2 players **Max** and **Min.**
- **Max** tries to maximize their score and **Min** tries to minimize the score



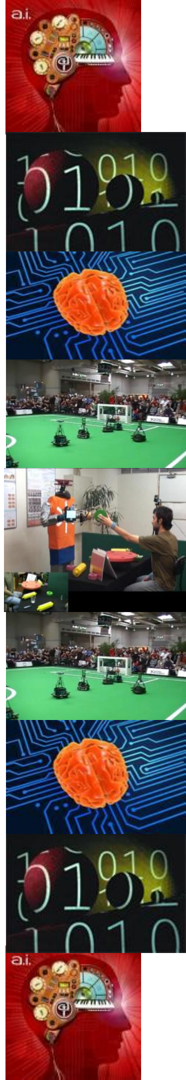
Minimax Algorithm

- Depending on the game intermediate nodes on the tree represent the states, arcs the moves and leaf nodes the score (utility) associated with that path.



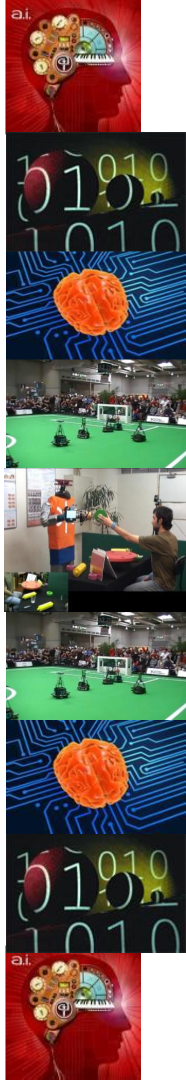
Minimax Algorithm

- The players alternate playing.
- So the first move (layer) may represent **Max** and the next **Min** until we get to the leaf nodes which carry the utility values.



Minimax Algorithm

- Utility values can be channelled back up to their parents.
- The **Max** player tries to maximize the minimum values at that position.
- The **Min** players tries to minimize the maximum values at that position.



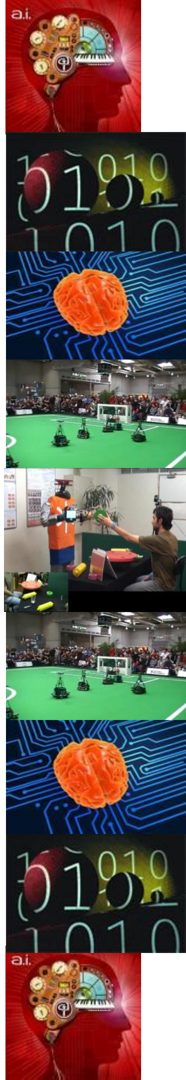
Minimax Algorithm

Algorithm 1. Minimax Algorithm

Repeat

1. If the limit of search has been reached, compute the static value of the current position relative to the appropriate player. Report the result.
2. Otherwise, if the level is a minimising level, use the minimax on the children of the current position. Report the minimum value of the results.
3. Otherwise, if the level is a maximising level, use the minimax on the children of the current position. Report the maximum of the results.

Until the entire tree is traversed

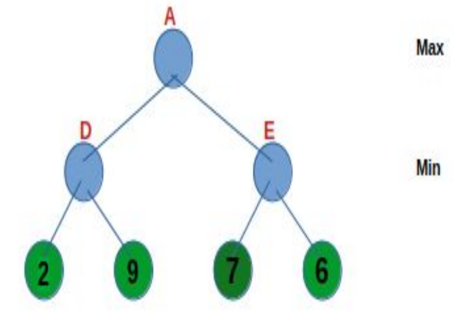


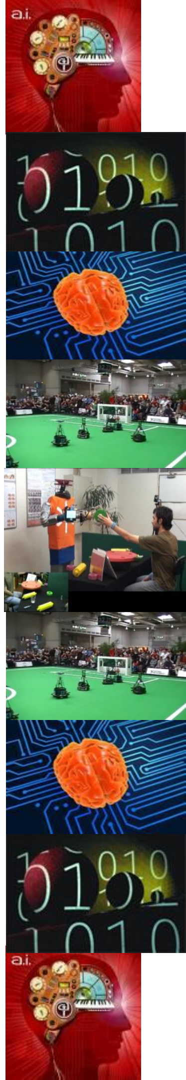
Minimax Algorithm

Example - We need to find the optimal path.

- Player **Max** and **Min**
- Assign default values Max (-infinity), Min (+infinity)
- **Max** plays followed by **Min**
- We follow Depth First Search … down to D.
- At D we minimize **D= min(+∞,2) =2** then we evaluate the other leaf node **D= min(2,9) =2.**

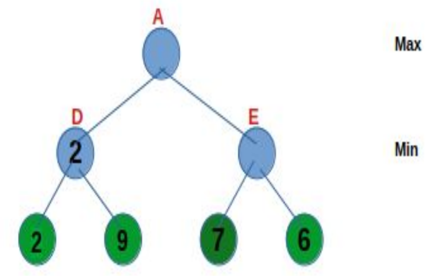
- Therefore the node D is set to 2 by the Min player.

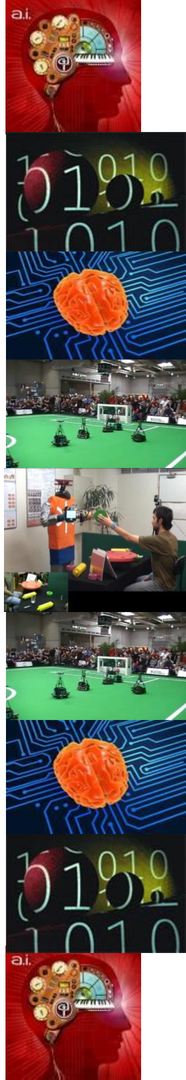




Minimax Algorithm

- The algorithm backtracks to **node A**
- Then to **node E** again its the Min player therefore we minimize.
- **D= min(+∞,7) =7 then min(7,6) = 6.**
- Thus the value of **node E** is evaluated to be 6

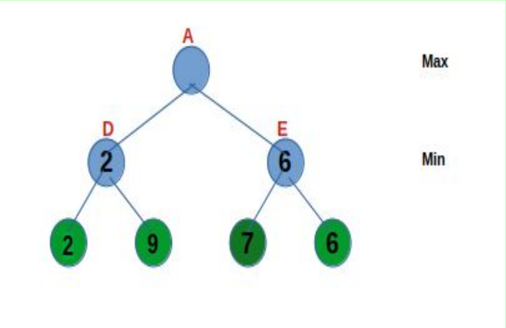




Minimax Algorithm

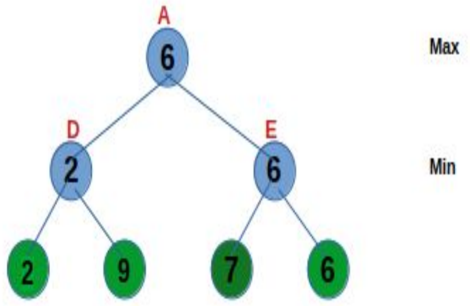
- The value of **node E** is set to be 6.

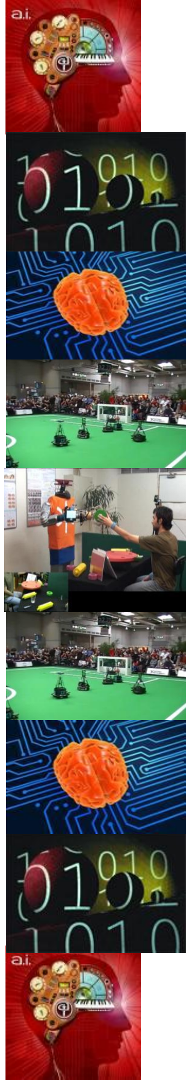
- At this stage the algorithm is moves to player **Max**



Minimax Algorithm

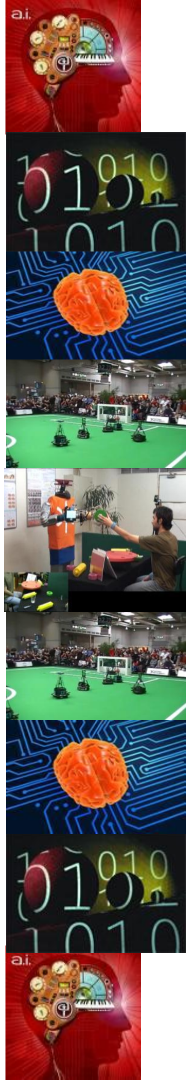
- The Max player perform maximization as follows:
- **A= max(-∞,2) = 2 then max(2,6) = 6.**
- From this example the Max player should play by choosing to go to **node E**
- The tree contains higher values than 6 yet A can only achieve utility value of 6.
- This shows that the Min player is playing optimally





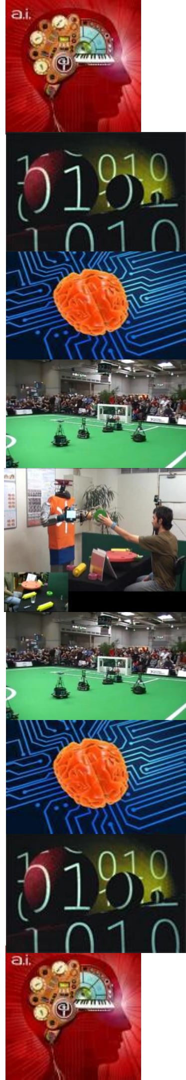
Minimax Algorithm- Advantages

- It considers all possible moves.
- Hence allows for the most optimal moves to be made.
- Highly accurate.



Minimax - Disadvantages

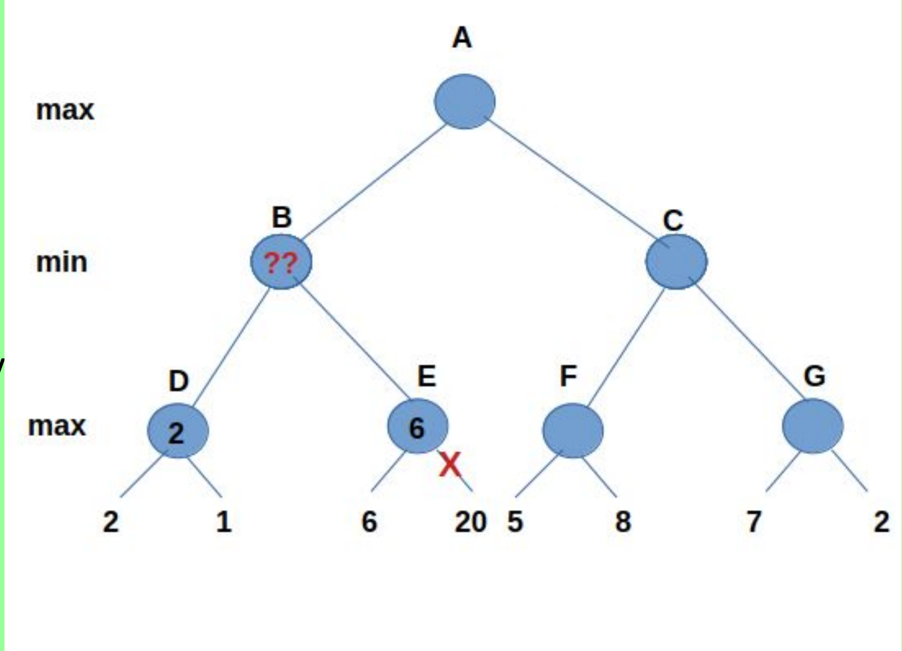
- The main drawback of minimax is that for complex games (e.g chess) that have a large branching factor it is very slow.
- Hence the use of Alpha-Beta Pruning.

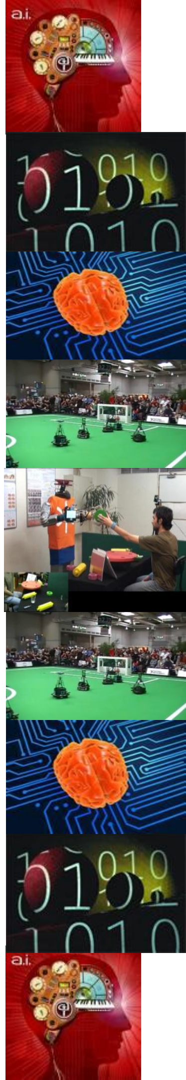


Alpha-Beta Pruning

Minimax Algorithm

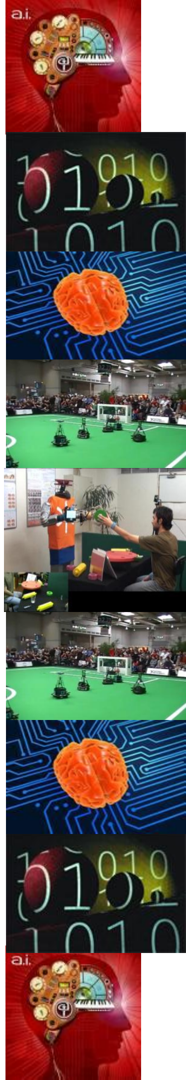
- At **node B** once **node E** has evaluated the left leaf node its clear that the value of **node B** will not be greater than 2. Thus its not necessary to evaluate the value of the right leaf of **node E.**
  What if 6 was 1 ??





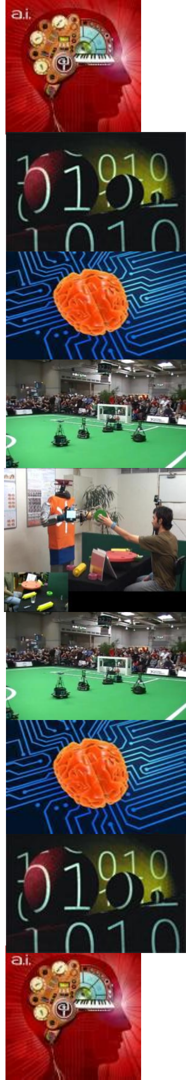
Alpha-Beta Pruning

- Game trees can be enormous.
- Some branches have no effect.
- Several branches can be removed.
- Pruning the branches reduces the time for the minimax algorithm.



Alpha Beta Algorithm

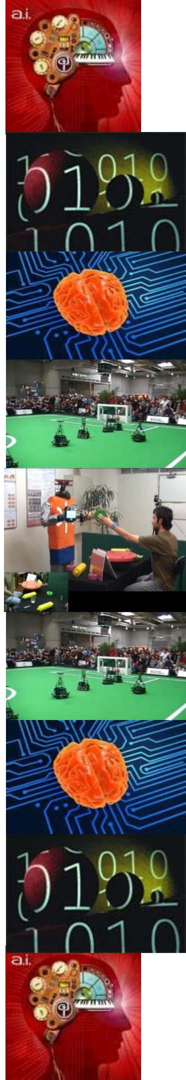
Two extra variables are assigned to every node in the tree (alpha & beta variables).



Alpha Beta Algorithm

<u>Alpha</u>

- This variable stores the best already explored option along a path to the root for the max layer.
- This is the maximum score the maximizer can receive.
- Only the maximizer can change this value.
- Its initial value is ***-infinity***.



Alpha Beta Algorithm

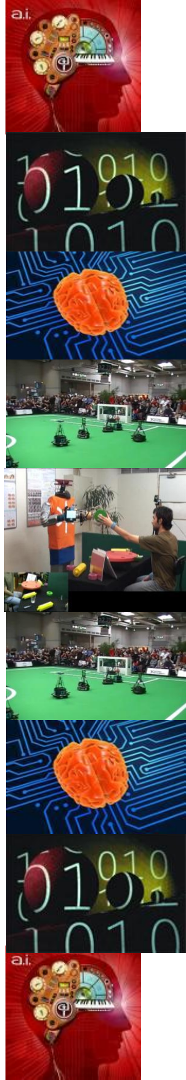
<u>Beta</u>

- This variable stores the best already explored option along the path to the root for the minimizing layer.

- This is the minimum score the minimizer can receive.

- Only the minimizer can change this value.

- It has an initial value of ***infinity***



Alpha Beta Algorithm

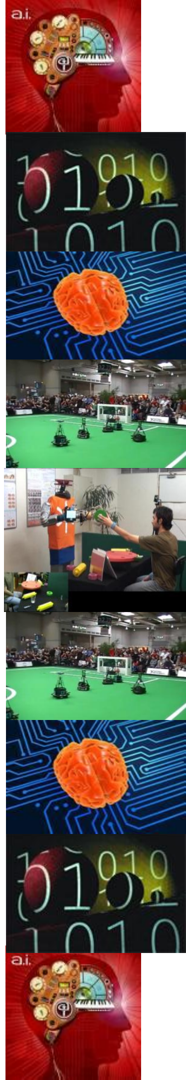
If α >= β we can prune the given branch

Alpha Beta Algorithm

***Algorithm 2**: Minimax Algorithm with Alpha-Beta Pruning*

Set alpha to -infinity and set beta to infinity
If the node is a leaf node return the value
If the node is a min node then
- For each of the children apply the minimax algorithm with alpha-beta pruning.
  - If the value returned by a child is less then beta set beta to this value
  - If at any stage beta is less than or equal to alpha do not examine any more children
- Return the value of beta

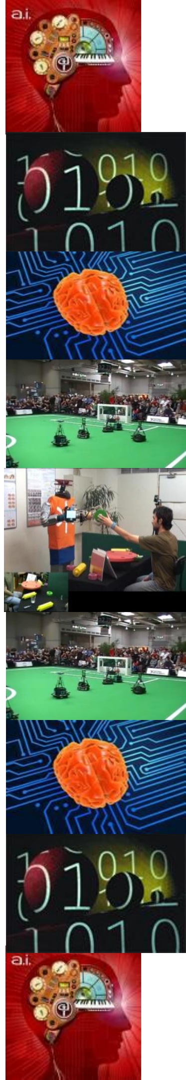
If the node is a max node
- For each of the children apply the minimax algorithm with alpha-beta pruning.
  - If the value returned by a child is greater then alpha set alpha to this value
  - If at any stage alpha is greater than or equal to beta do not examine any more children
- Return the value of alpha



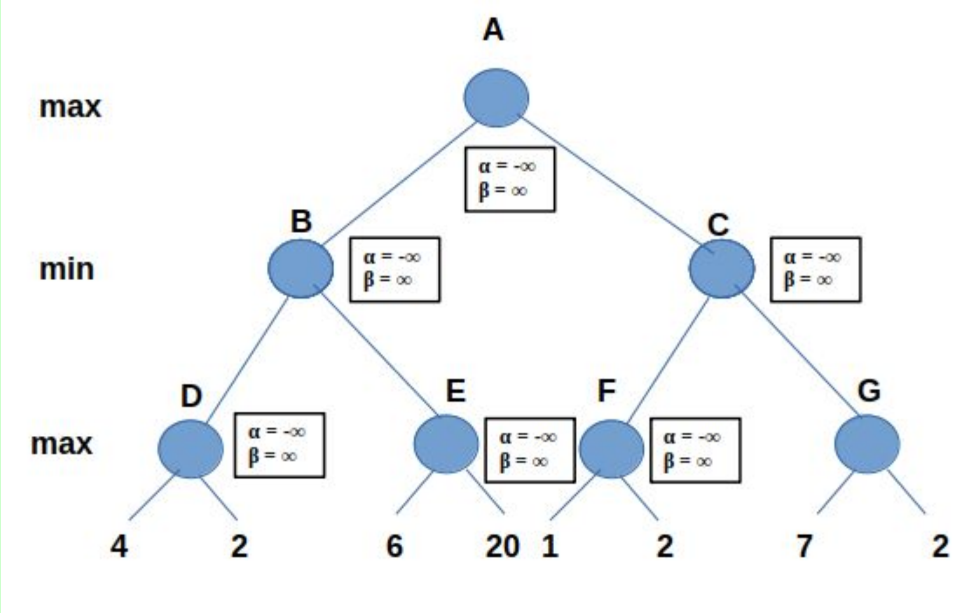
Alpha Beta Algorithm

- Pruning is not guaranteed to happen.

- Depending on the order of the distribution of the values.



Alpha Beta Algorithm



Alpha Beta Algorithm

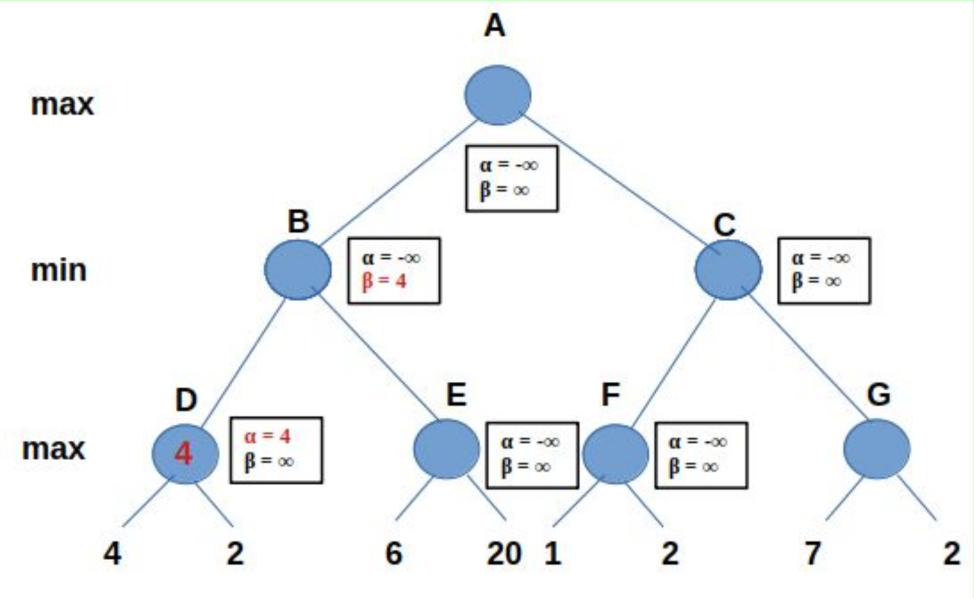
At **<u>node D</u>**

D set to 4

α = 4

At **<u>node B</u>**

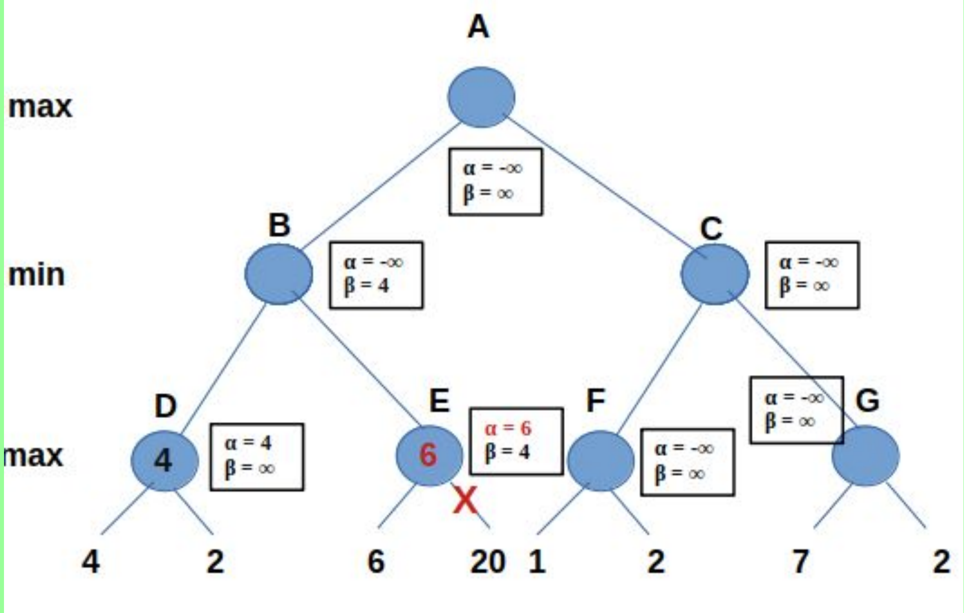
β = 4 (best min value at thus far from node D



Alpha Beta Algorithm

β (4) is passed down to **node E** from B.
@ **<u>node E</u>** the left Leaf provides 6 to E & **α**.
<u>**α** > **β** we prune</u>

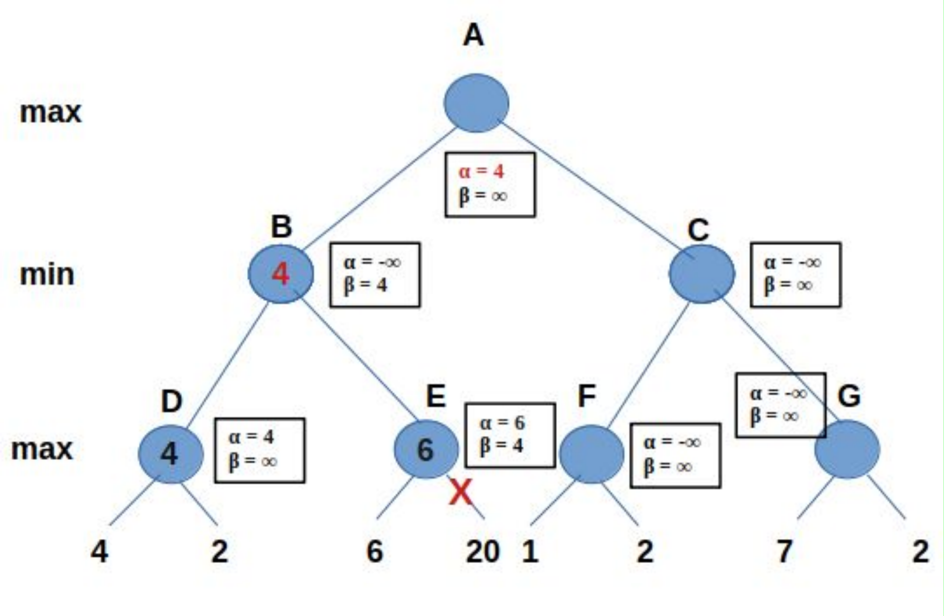
α update to 20 ???

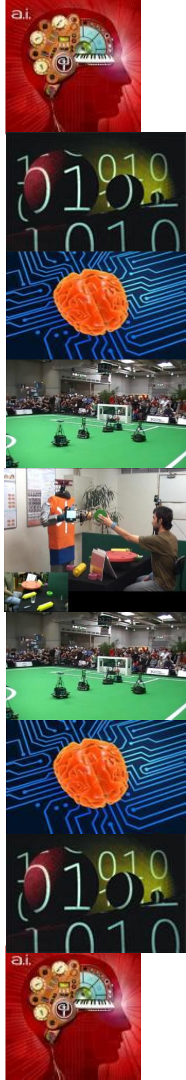


Alpha Beta Algorithm

@ **node A**

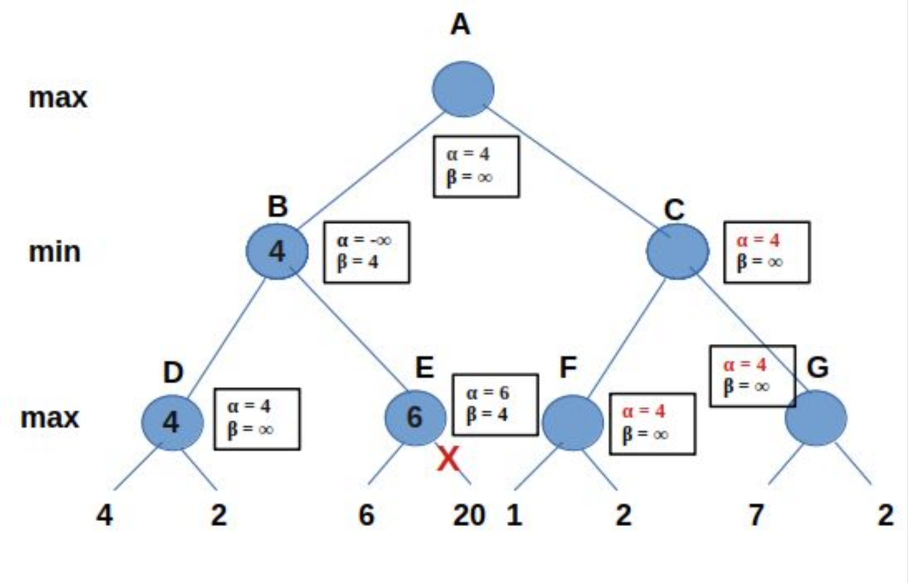
α is updated to 4 after backtracking from **node B.**

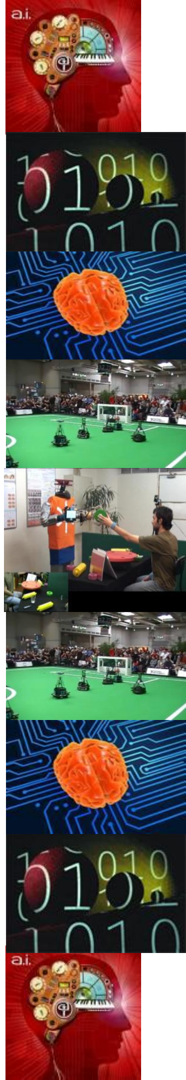




Alpha Beta Algorithm

- **α** values are cascaded down on the right of node A.
- Dfs is applied till **node F.**





Alpha Beta Algorithm

- @ **<u>node F</u>**

α is not update

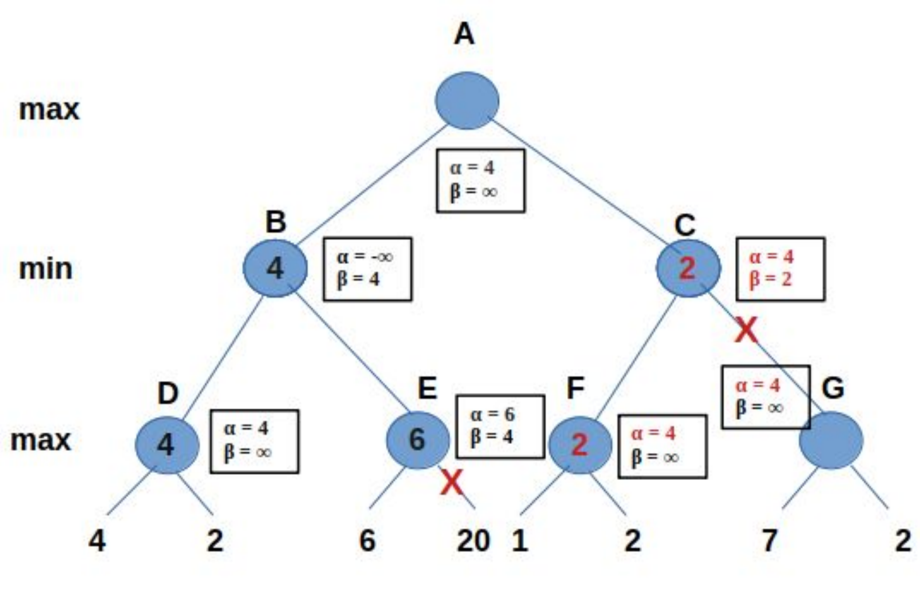
Why not ??

Node F =2.

- @**<u>node C</u>**

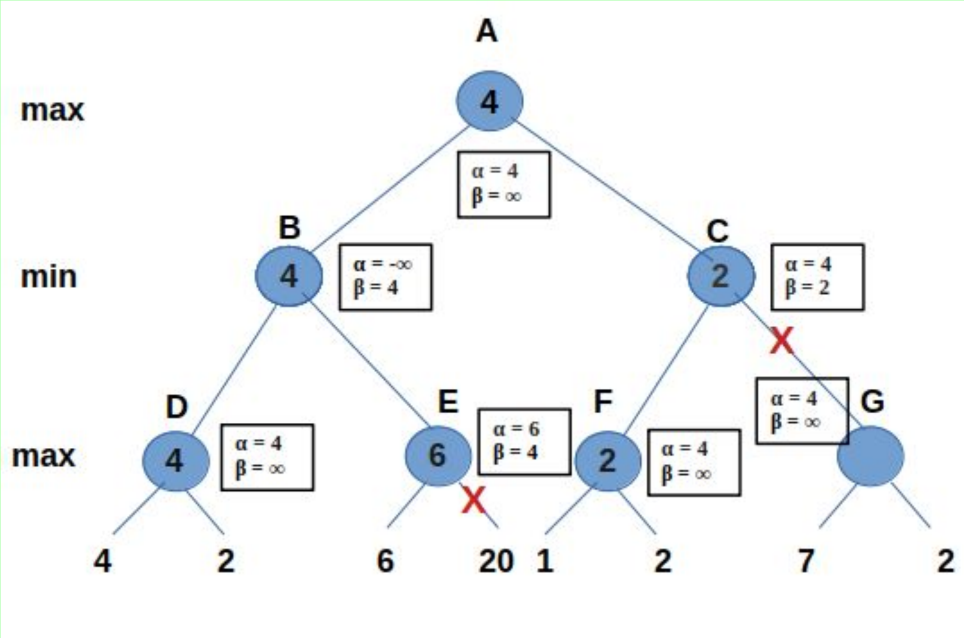
β is updated.

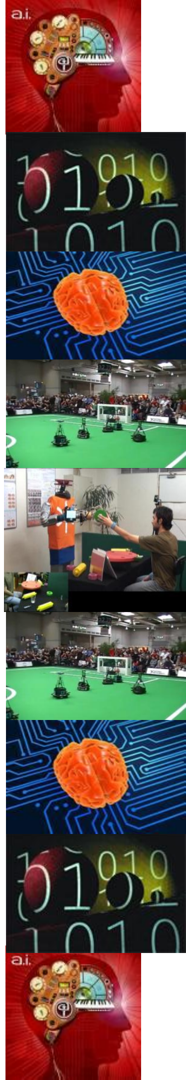
<u>α > β</u> <u>we prune</u>



Alpha Beta Algorithm

- @ **node A**
  Value is updated

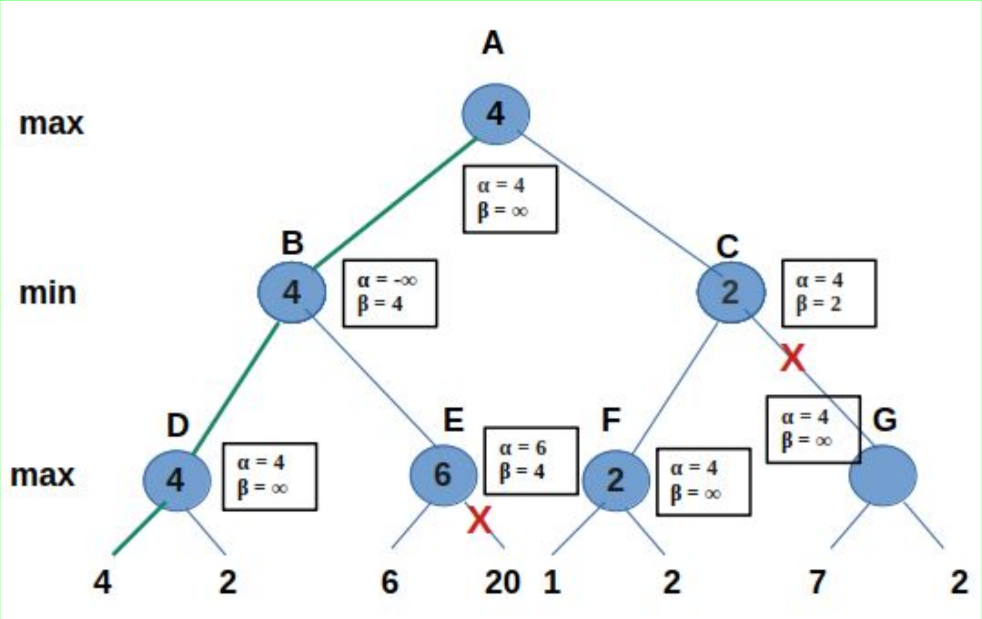


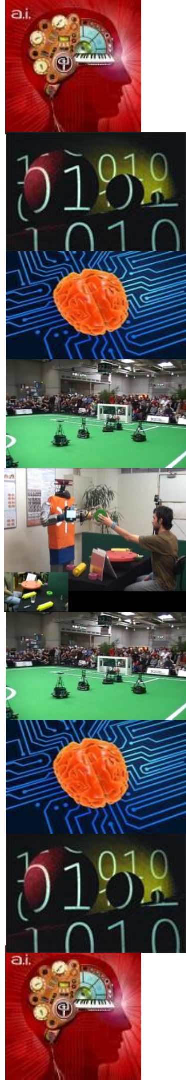


Alpha Beta Algorithm

Optimal path

A-B-D-4





QUESTIONS