Minmax algorithm

1. First import the library (import math) because it depends on math.
2. There will be two player first player use min and second person use max.
3. Make a leaf node in which values are stored and name it as values.
4. Use math.log function and give length of values and a base 2.
5. Make a function and pass five parameters (currnode,depthnode,minturn,value, treedepth).
6. Use if condition if currdepth equal to treedepth it will return value of node index.
7. Again use if else condition make first condition for minturn in which it return (currnode+1 ,nodeindex\*2,False,value,treedepth), (currdept+1,nodeindex\*2+1,False,values,treedepth).

* currnode+1 it will tell current position if it is empty it will move to next with help of +1.
* depthnode\*2 it will tell the depth of node in row in even and for the next time (nodeindex\*2+1) it will check for odd node because min is false.
* pass the value of min as false so it will be zero.
* pass the values (leaf nodes).
* pass the depth of tree and it will take the depth from level where we have use math.log.

1. Do it again use if else condition make first condition for maxtturn in which it return (currnode+1 ,nodeindex\*2,True,value,treedepth), (currdept+1,nodeindex\*2+1,True,values,treedepth)

* currnode+1 it will tell current position if it is empty it will move to next with help of +1
* depthnode\*2 it will tell the depth of node in row in even and for the next time(nodeindex\*2+1) it will check for odd node because max is true
* pass the value of max as true so it will be one
* pass the values (leaf nodes)
* pass the depth of tree and it will take the depth from level where we have use math.log

1. Call the function and give value of 5 parameter which you have passed.
2. Print the result.