n-order & level-order not possible 4 due to both traversals doern't provide info regarding labout the structure of the tree to uniquely construct ? t post order & Breorder male ( 1 + 2 + 200 ) The contract Both have the some reason that Not possible to uniquely Edenti a tree Because Both traversal do not provide enough info. about the Structure of the tree to unquely MINITED CONSTRUCTS THE SUIT MATERIA agriculture to the total the total various

Date
in point (a) A point (b) it is possible to
En pout (a) A pout (b) it is possible to consignely identify a tree for the given traversal Combinations
are - @ postorder & level-order
(b) Fur ordere & bre-order
a) -> possible because level-prodey traversal
possible because level-brder traversal
the order ( in which Noder and
placed at call (evel)
postrorder provides pre order (en www.
moder are visited after their (wid)
so, these two reason Allasing us to secons.
Purorder - & fore order
) -> possible because Furorder traversas
ave Visted [left aubtree, root, rèque son
and pre-order provider the root followed by left & regul publice
allowing us to reconstruct the tree
anquely

Prodocate to a
Pswedo code to construct tree
The state of the s
function coul tree [1]
function cous_tree [bost order, level order]:
empty r  return None
return None alling (B)
as assess with the same some
Noot value - last
root value = last element of post order
root = new Tree Node (root Value)
(2000) 4220 Maria
(2000) TELLE LOUR DESIGN DIES 19 MARCE
and the fundament of the contract of the contr
index = index of root value in level order
Lections of a state - least to Page 7
left level order = level order [o; endex]
right level order = 1evel order [ Ender +1 %]
a desile state of a series of
reft Postorder = dements of post order that
aue in lettlevel order
rightPost order= elements of post order
that are en right levelord
did and the total and promise
moot. left = constree [reftPostorder,
left Level order)
700 t. riquit =
root. régut = cons-tree (régutPostorden,
right level order
return moot
Spiral

function Com-tree (inorder, preorder).

If inorder is empty or preorder is empty.

Teturn None

root Value = First element of brearder root = new TreeNode (root Value)

Endex = Endex of rootvalue en Enorder

reghtinorder = enorder [orindex]
reghtinorder = enorder [endex + 1:]

left Preorder = elements of preorder that a In left inorder

regulapreorder = plements en preorder that

root. left = 1000-tree (leftinorder, left f

noot right = constree (régutinorder, rights

return root