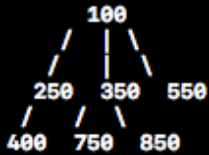


TIMING

LCRSTREE

TEST CASE FOR LCRSTREE

TREE:



----- Height 3 the time for LCRSTREE PREORDER ----- :

PREORDER is = 100 250 400 350 750 850 550

Time taken is = 0.032

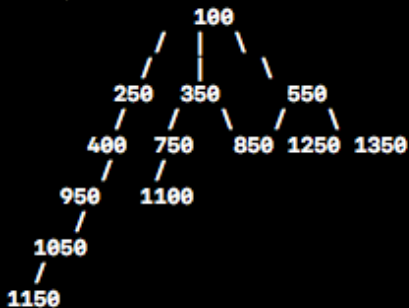
----- Height 3 the time for LCRSTREE POSTORDER ----- :

POSTORDER is = 400 250 750 850 350 550 100

Time taken is = 0.035

TEST CASE 2 FOR LCRSTREE

TREE:



----- Height 6 the time for LCRSTREE PREORDER ----- :

PREORDER is = 100 250 400 950 1050 1150 350 750 1100 1103 850 550 1250 1350

Time taken is = 0.005

----- Height 6 the time for LCRSTREE POSTORDER ----- :

POSTORDER is = 1150 1050 950 400 250 1103 1100 750 850 350 1250 1350 550 100

Time taken is = 0.005

- Leftmost-child right sibling implementation's timing ;
 - Test Case 1
 - We can gather that preorder and postorder times has almost no difference. Postorder takes slightly more time because it has to go to that node then do it over and over again till everything is set but for the preorder it basically does node->next; that's I believe the preorder time is $O(n)$ and for postorder it should be $O(n^2)$ for recursively going through the tree but at the end since nodes takes so little memory from one to another the difference is incredibly small.
 - Test Case 2
 - We can gather that preorder and postorder times has no difference. Postorder should take more time but since this tree grows to one side larger than others it takes more time for preorder to do the ordering that's why that small difference in test case 1 is gone in test case 2.

LOCTREE

```
TEST CASE FOR LOCTREE
TREE:
      100
     /  |  \
    250 350 550
   /  /  \  \
  400 750 850
----- Height 3 the time for LOCTREE PREORDER ----- :
PREORDER is = 100 250 400 350 750 850 550
Time taken is = 0.006
----- Height 3 the time for LOCTREE POSTORDER ----- :
POSTORDER is = 400 250 750 850 350 550 100
Time taken is = 0.003

TEST CASE 2 FOR LOCTREE
TREE:
      100
     /  |  \
    250 350 550
   /  /  \  \
  400 750 850 1250 1350
 /  \
950 1100
/
1050
/
1150
----- Height 6 the time for LOCTREE PREORDER ----- :
PREORDER is = 100 250 400 950 1050 1150 350 750 1100 1103 850 550 1250 1350
Time taken is = 0.012
----- Height 6 the time for LOCTREE POSTORDER ----- :
POSTORDER is = 1150 1050 950 400 250 1103 1100 750 850 350 1250 1350 550 100
Time taken is = 0.005

Program ended with exit code: 0
```

- List of Children implementation's timing;
 - Test Case 1
 - For the first case preorder takes double the time of postorder, I believe it is because for postorder we go to the noleaf node and then make our way up so we don't have make the effort of going up and down, I think that's exactly why preorder takes double the time for ordering. So my conclusion is that $O(n)$ for postorder and $O(2n)$ for the preorder.
 - Test Case 2
 - For the second case preorder takes more than double the time of postorder which is not a surprise because as I explained for the first test case postorder makes only one way up on the other hand preorder makes ups and downs though the tree.