IN-ORDER TRAVERSAL

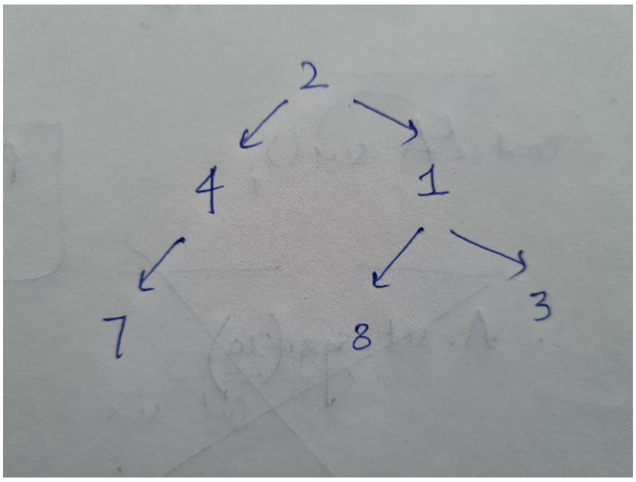
- Follows the L->**N**->R rule
- Ans: 7 4 2 8 1

PRE-ORDER TRAVERSAL

- Follows the **N**->L-> R rule
- Ans: 2 4 7 1 8 3

POST-ORDER TRAVERSAL

- Follows the L ->R->**N** rule
- Ans: 7 4 8 3 1 2



Following is the execution of the in-order traversal method for a binary tree -

1. Start from the root. Keep on going left until you reach a node with no child on its left. Start from 2 -> 4 -> 7 -> stop here as node 7 has no left child.
2. Now execute N for node 7 i.e. print node 7.
3. Now R i.e. check right for node 7. Here 7 has no right child.
4. As LNR has completed for 7, now move up, that is, go to node 4.
5. As; we have already checked L for node 4 so now we can simply execute N for it and thus print it.
6. Now check R. R empty. LNR completed for node 4. So move up to reach node 2.
7. As; we have already checked L for node 2 so now we execute N and print it.
8. Now check the right child of node 2 thus reaching node 1.
9. Check left of 1 thus reaching 8.
10. Check left of 8. No left child, so execute N and print 8.
11. Go to the right of 8. No child on right. LNR completed for node 8 so go up, reaching 1
12. L already executed for 1. So execute N and print 1.
13. Now check for R for 1 thus reaching 3.
14. Now check for L for 3. No child on the left of 3. Execute N and thus print 3.
15. Go to the right of 3. No child on the right of 3, LNR completed, go up, reaching 1.
16. We have already done LNR for 1. So go up, reaching 2.
17. Execution ends as there is no node above node 2 (parent node). This implies that LNR has been completed for every node