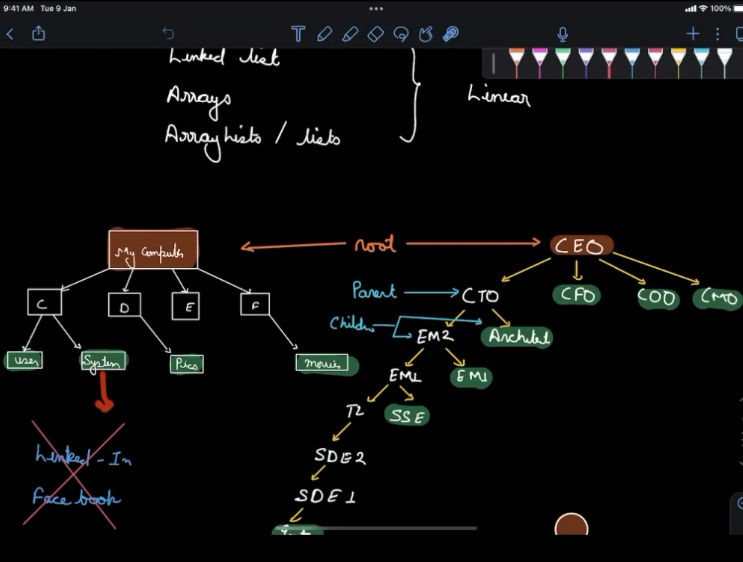
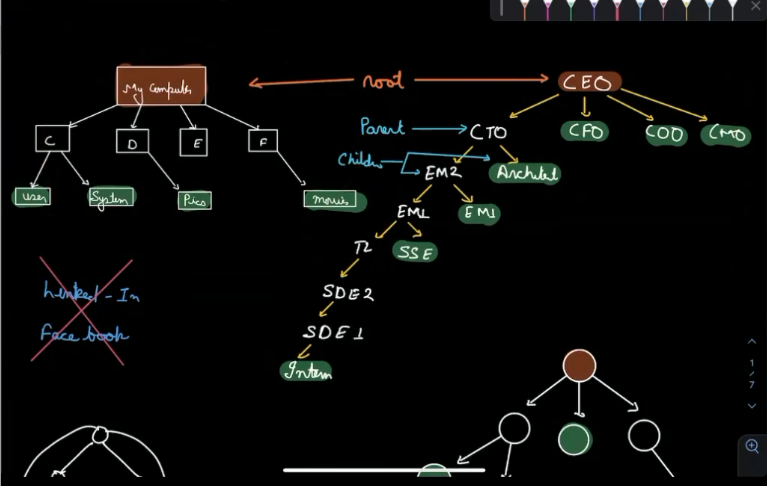
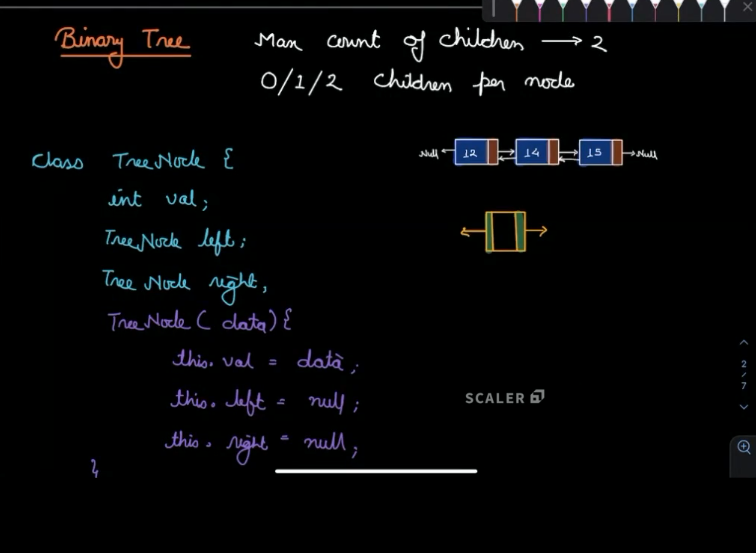
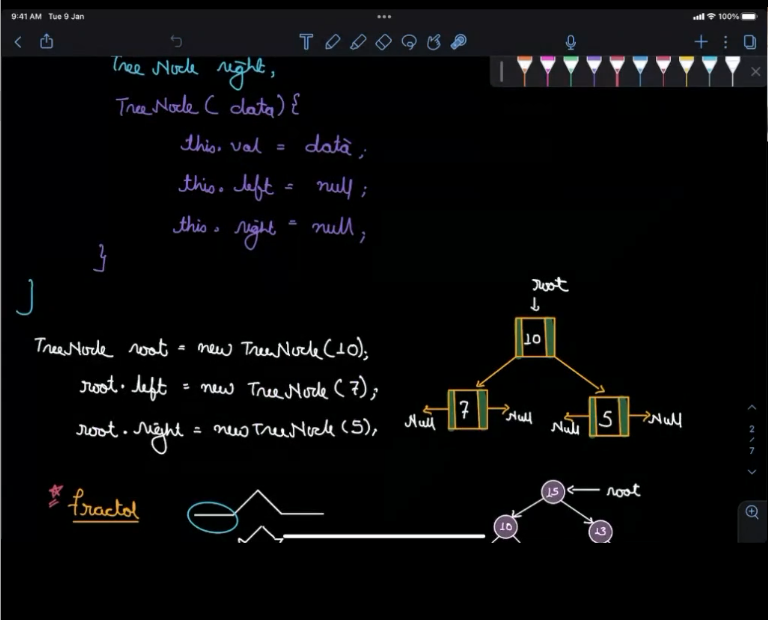
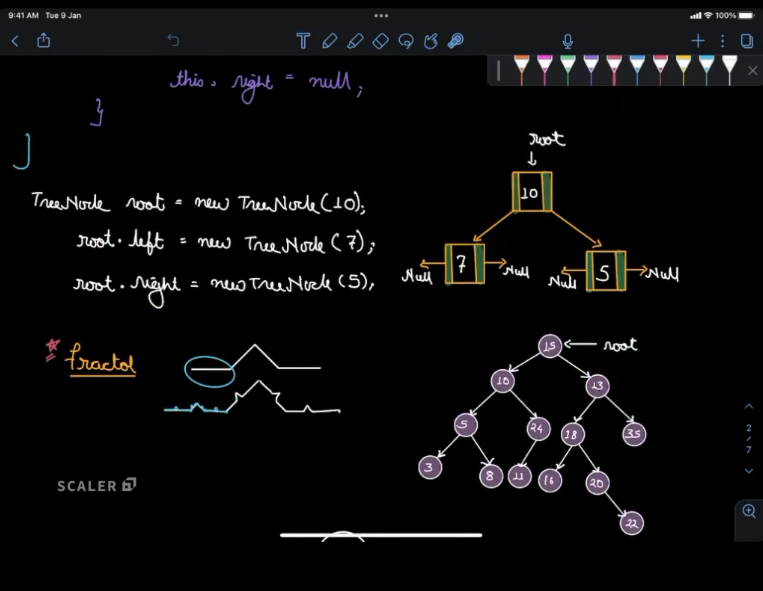
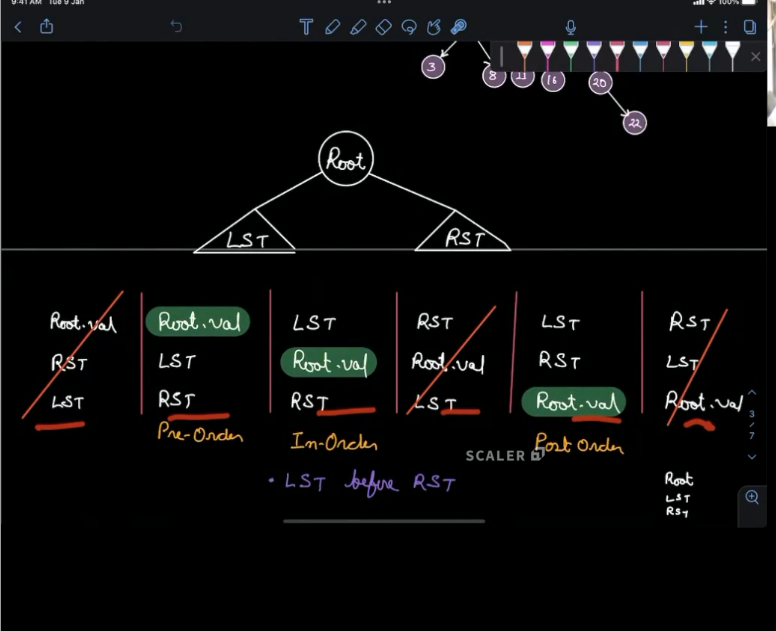
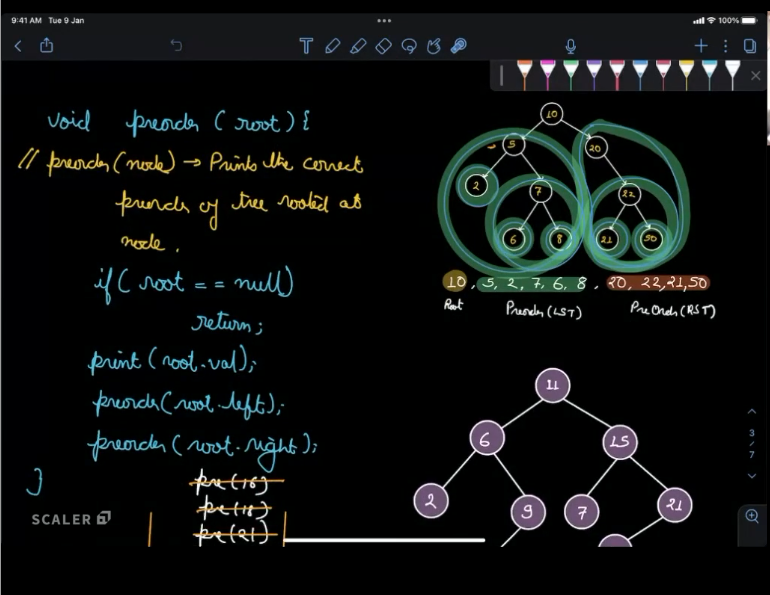
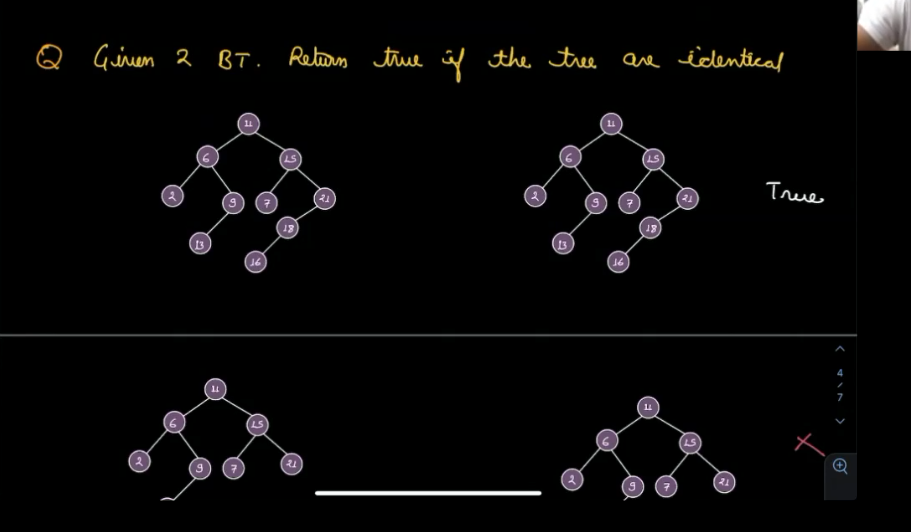
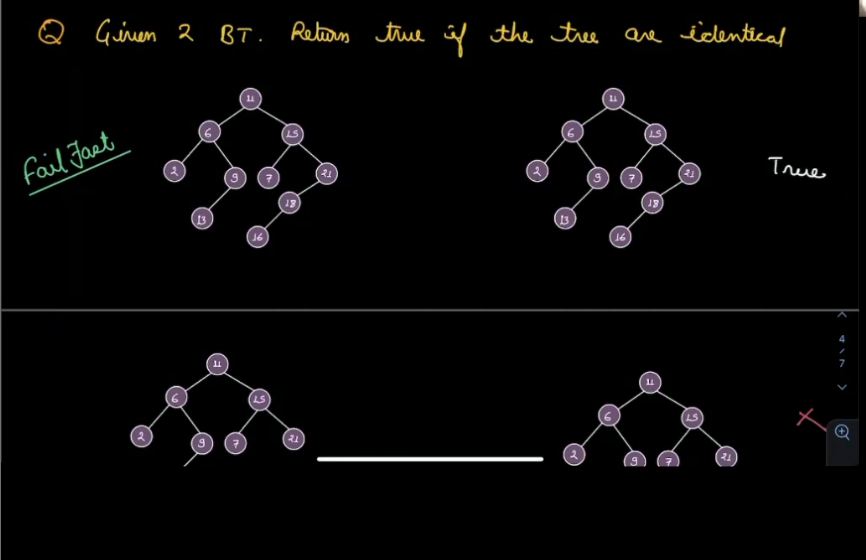
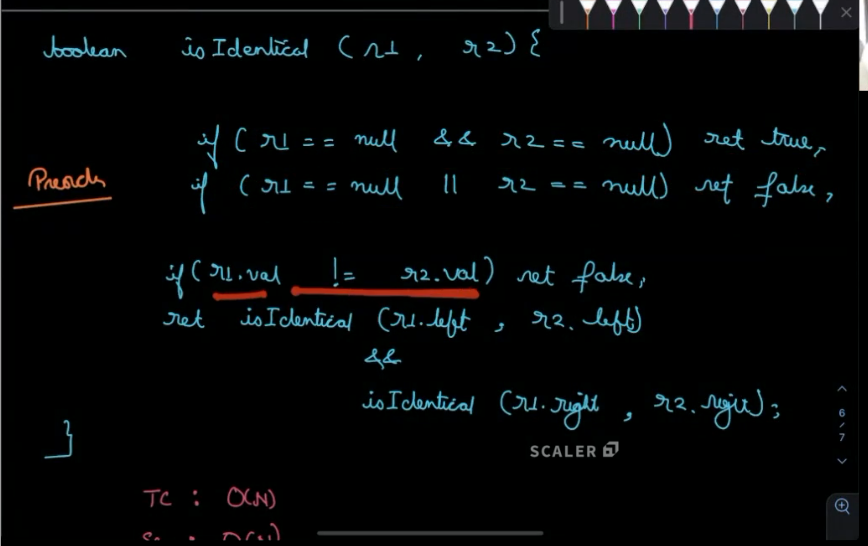
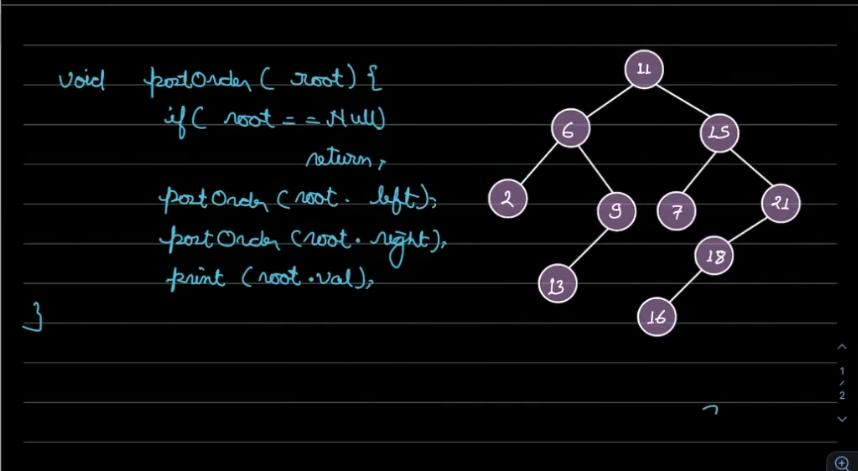
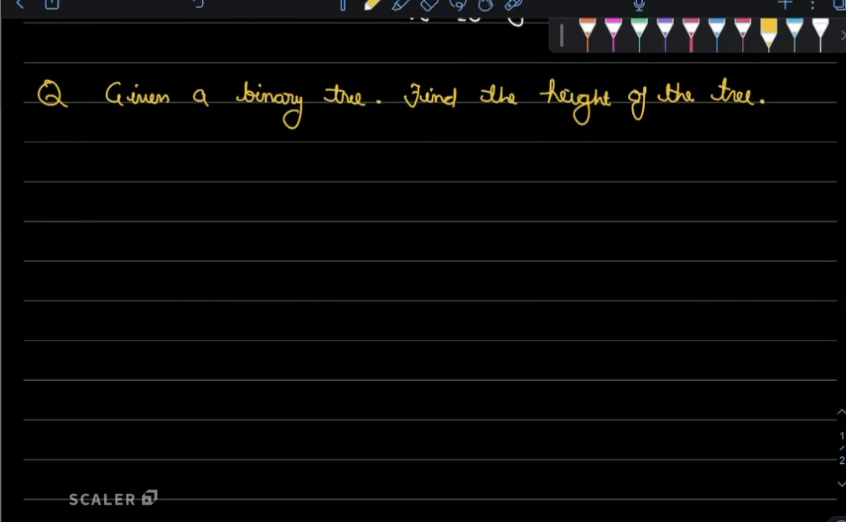
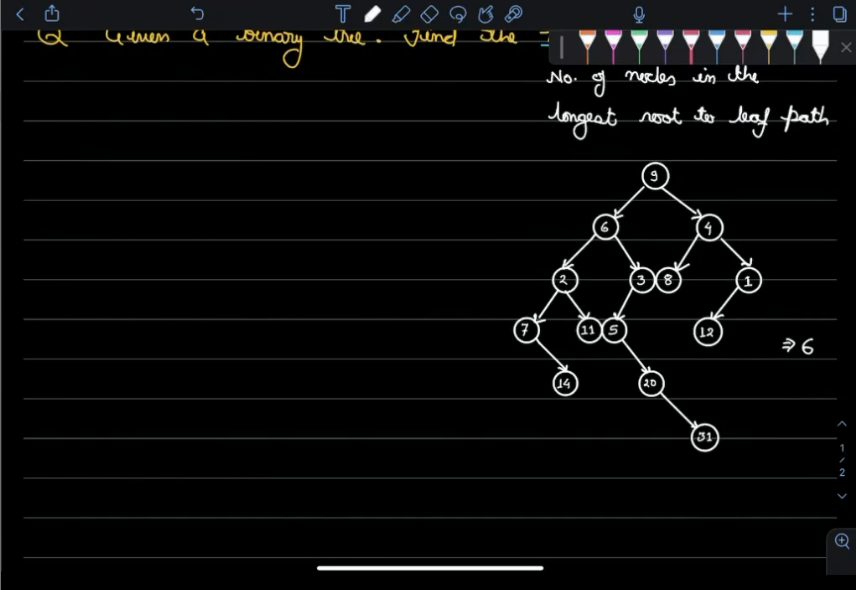
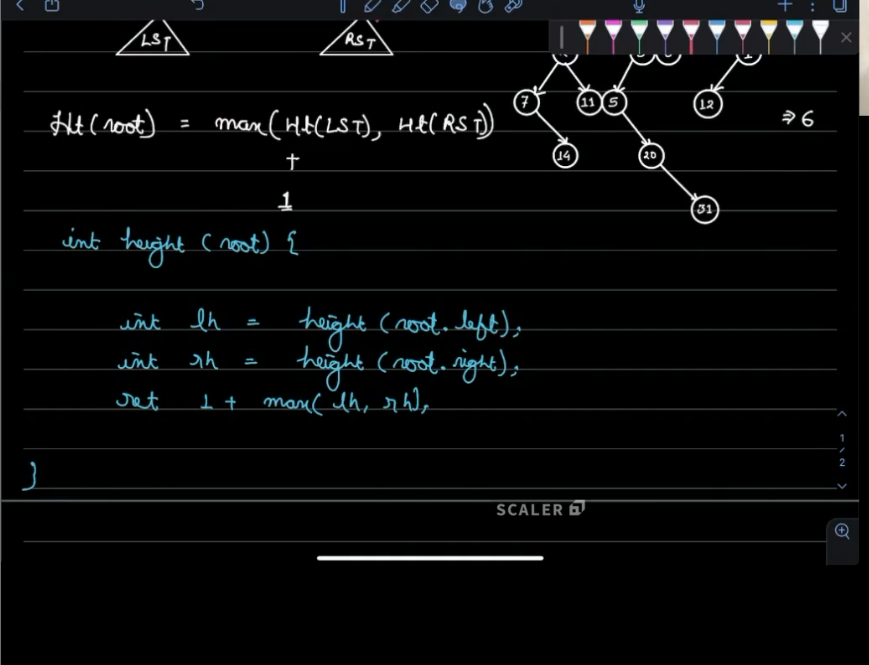
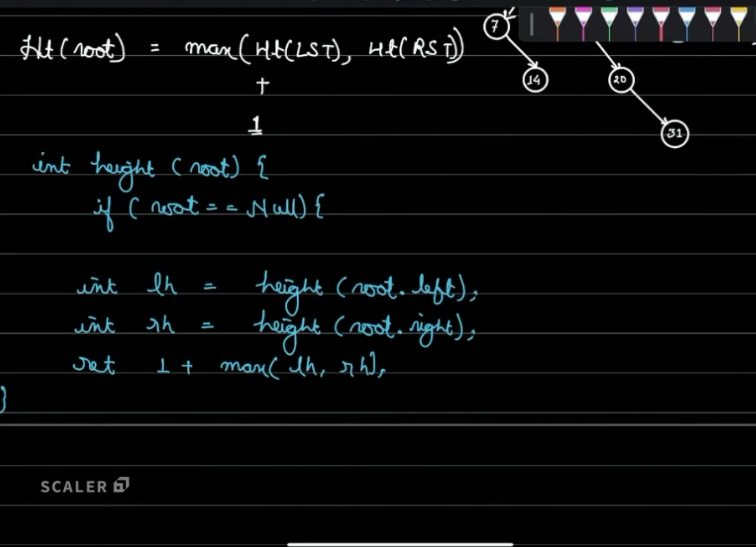
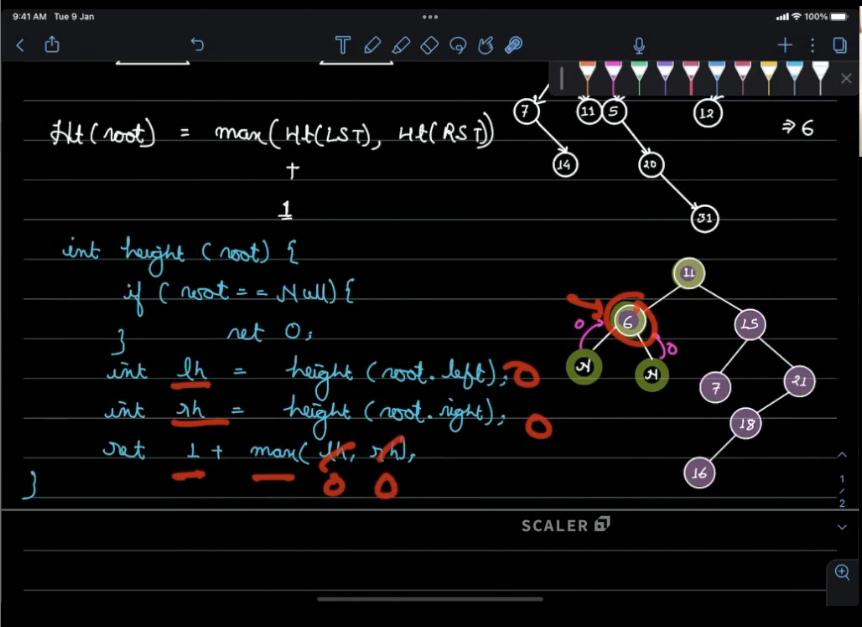
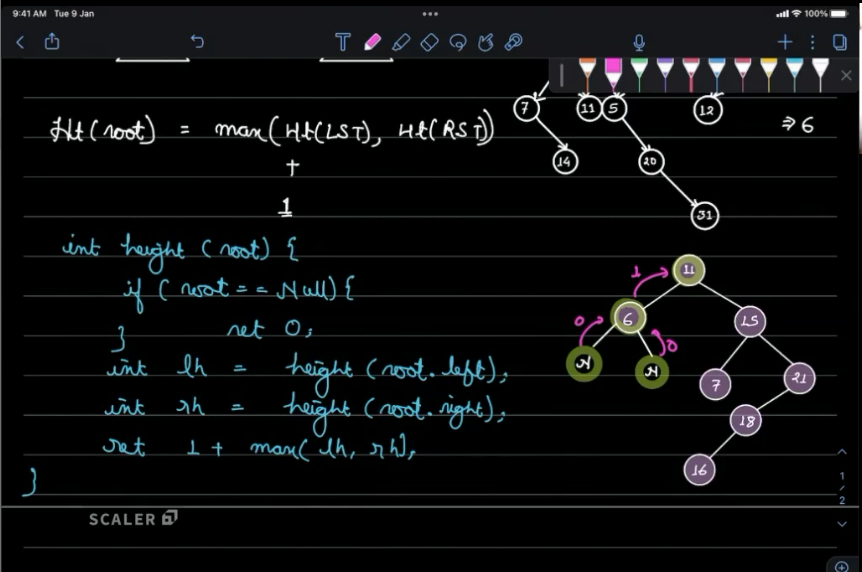
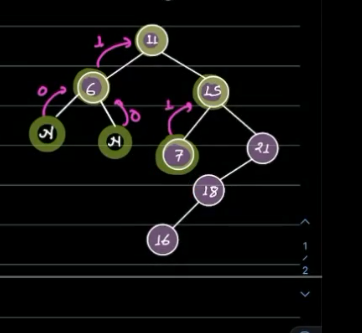
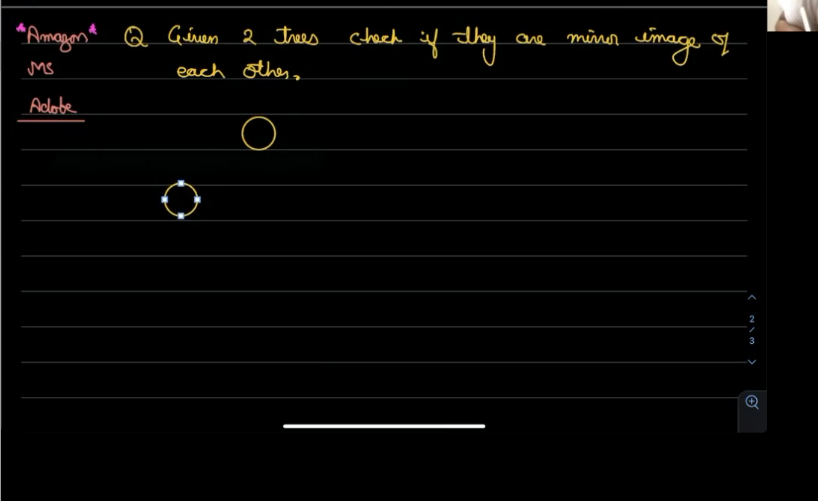
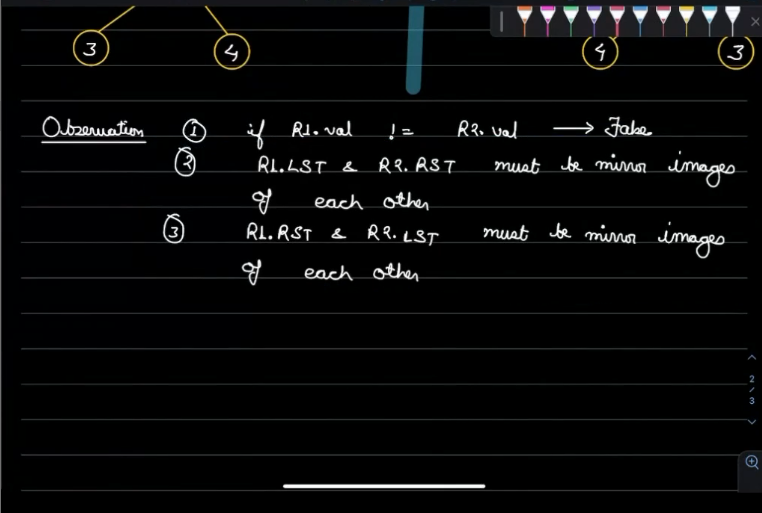
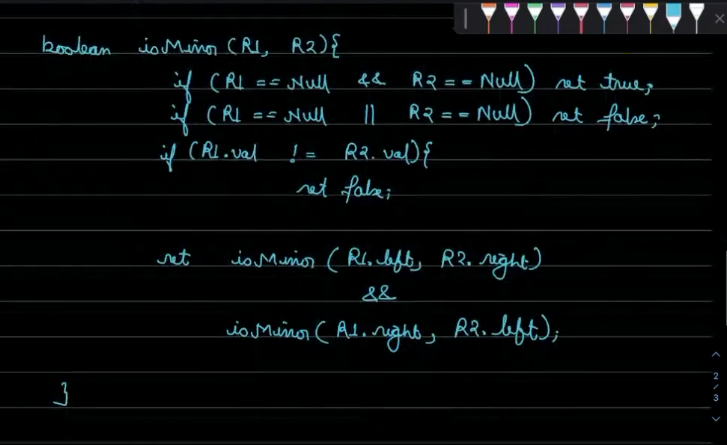
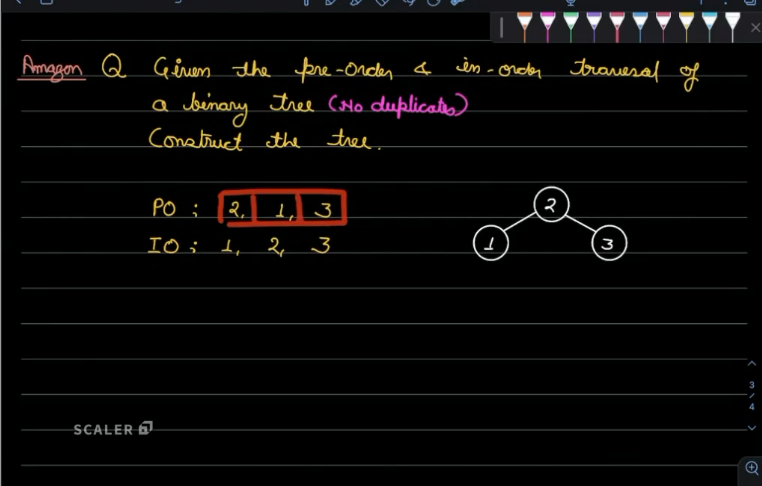
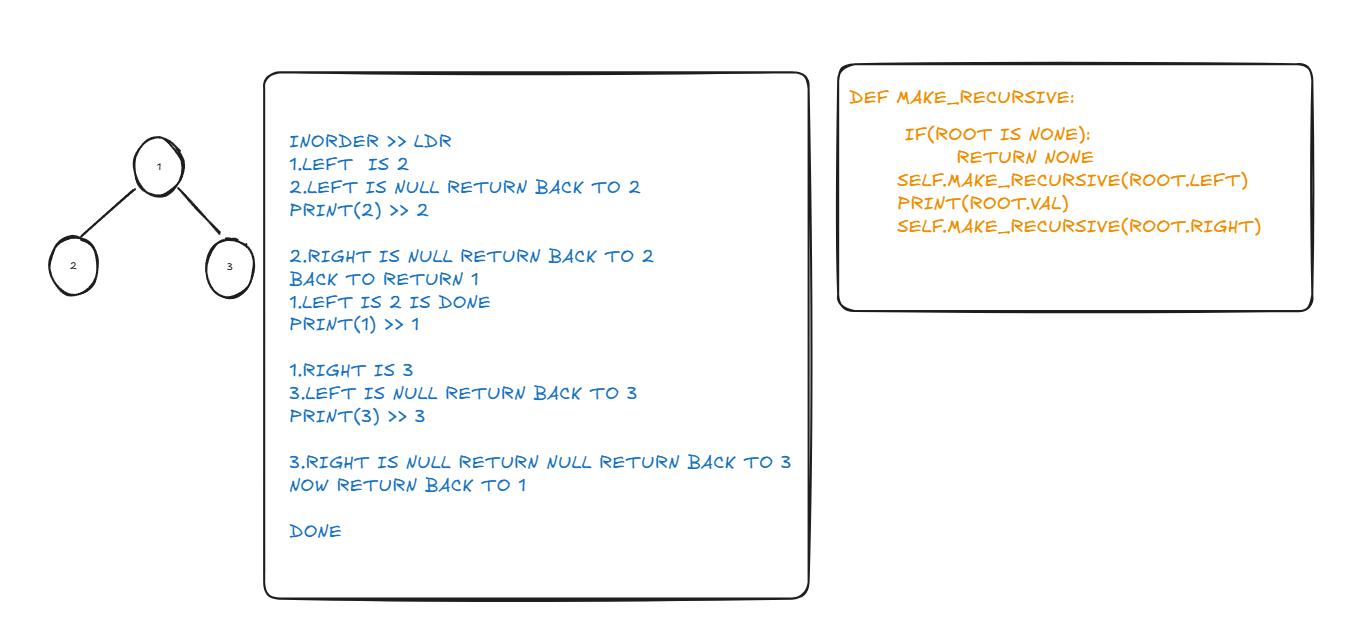
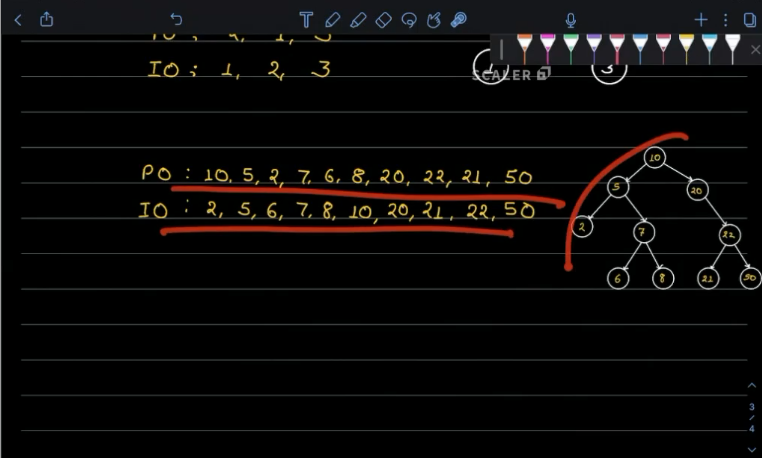
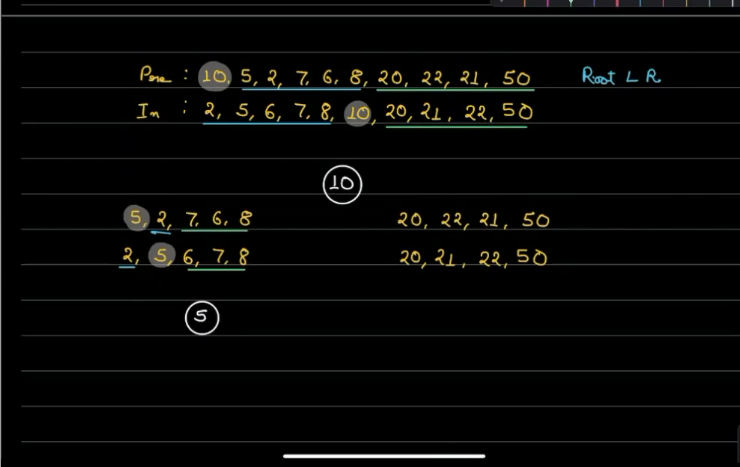
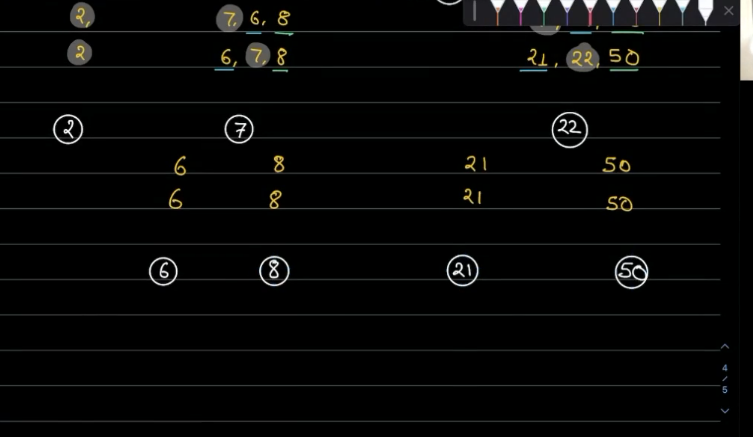
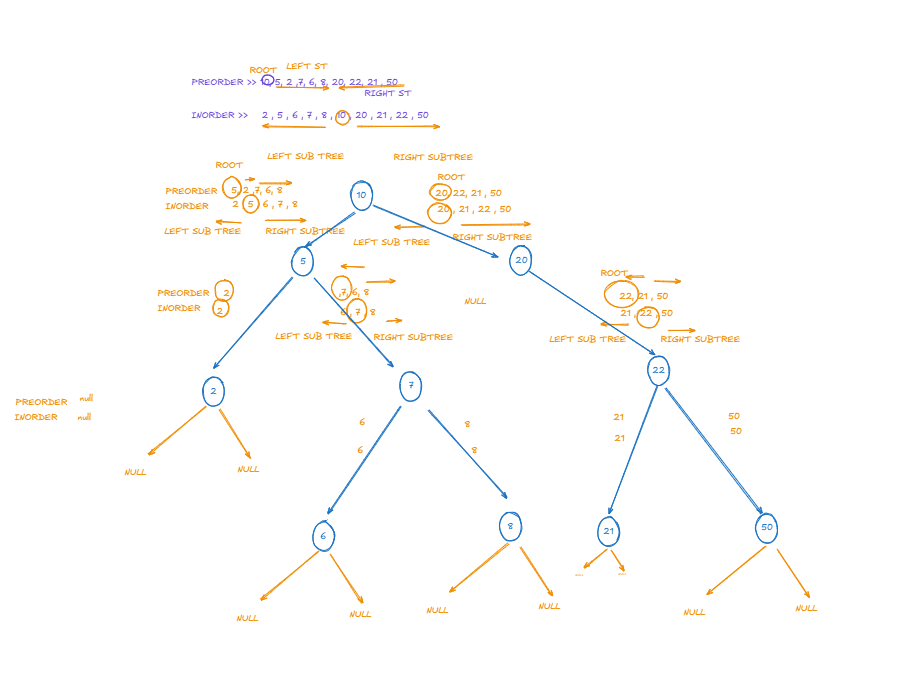
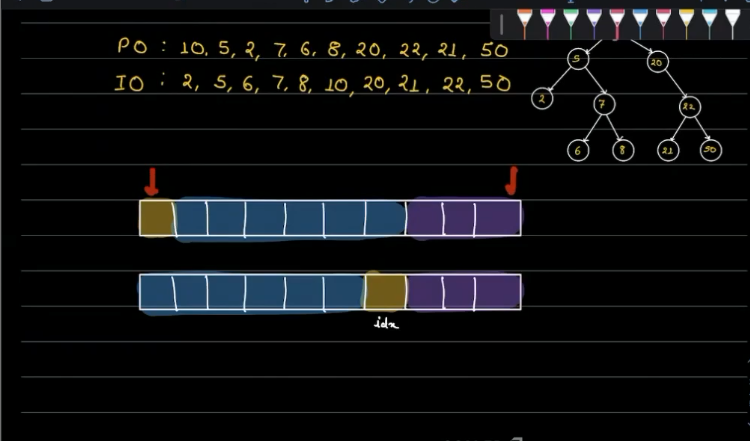
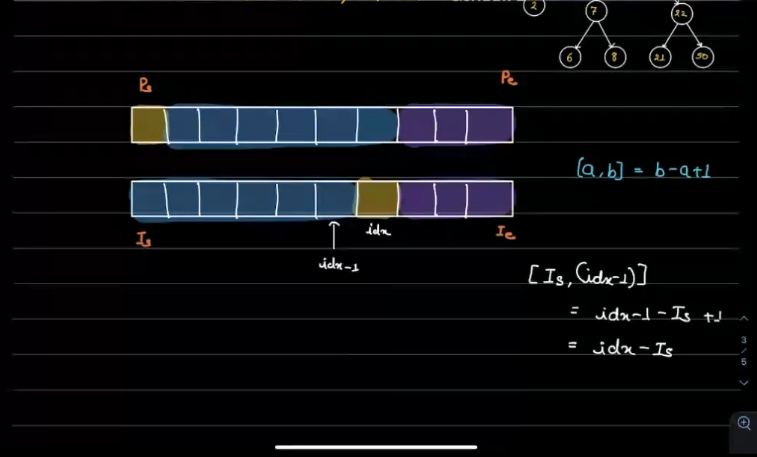
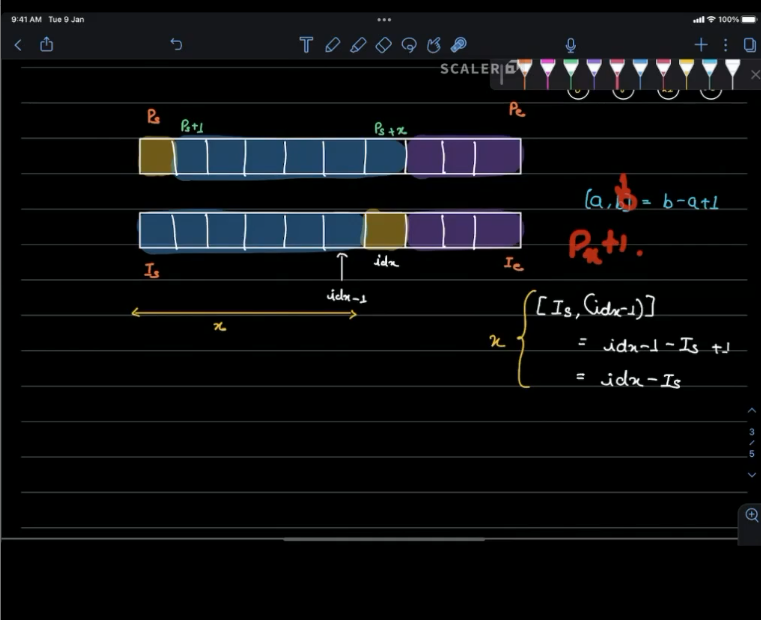
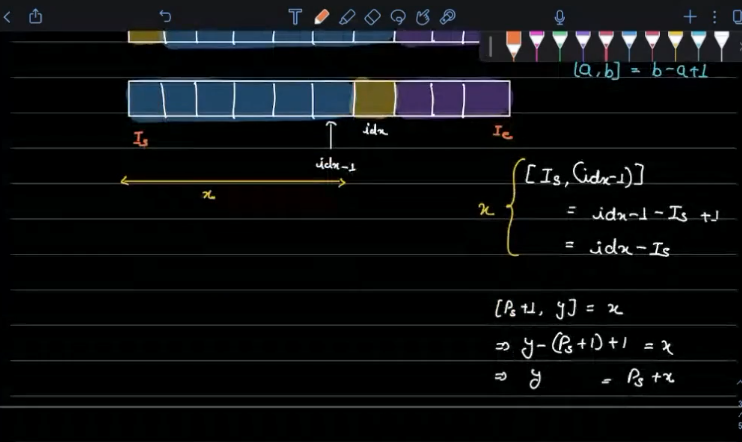
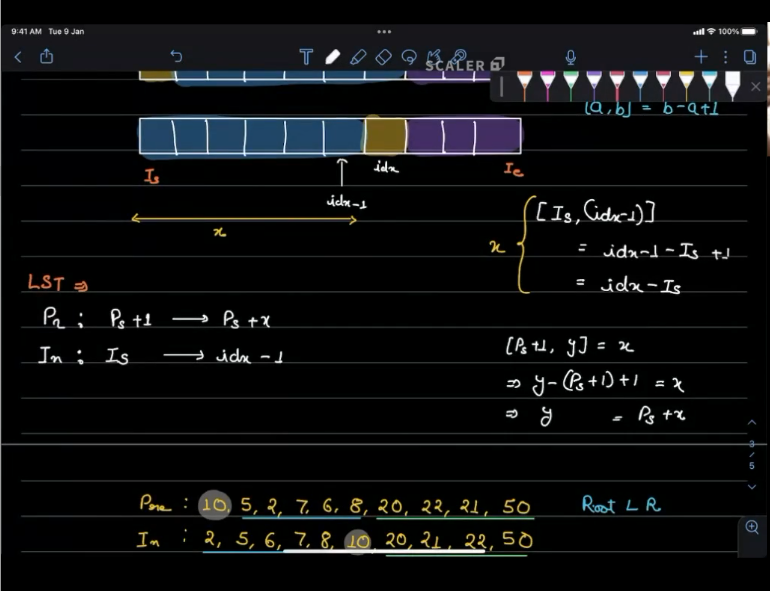
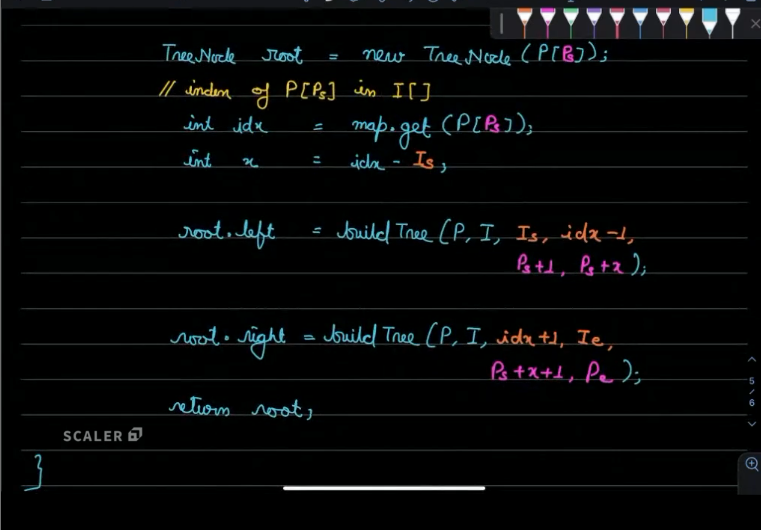
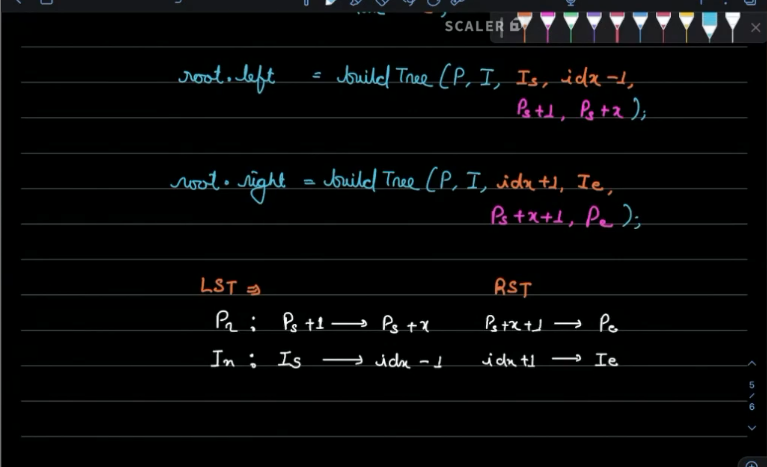
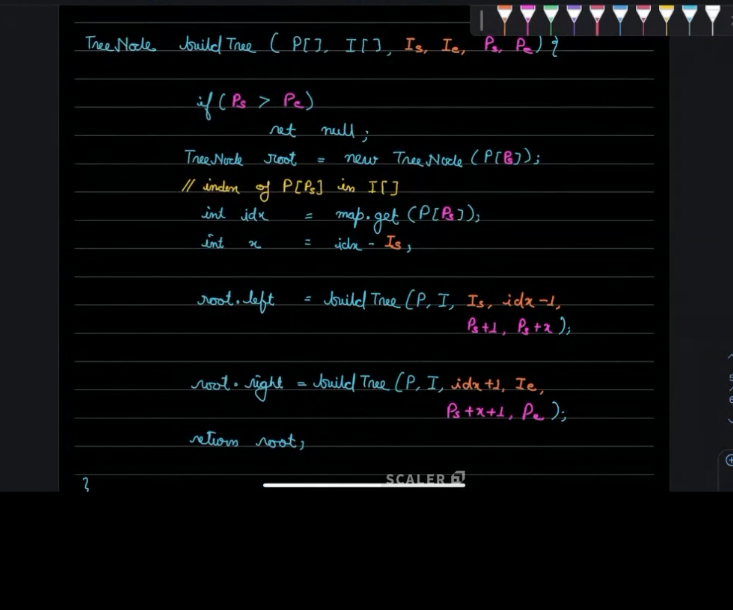
1. **LINEAR ARRANGEMENT OF DATA**
2. ****
3. **HEIRARCHY LIKE JOB ROLES**
4. **CONTAINER WHICH ACCESSING MEMORY**
5. ****
6. **ANY LEVEL CAN CONNECT WITH ANY LEVEL**
7. **SUBSET OF GRAPH**
8. **TREE**
9. **ROOT NODE AND LEAF NODE**
10. **HTML TREE**
11. ****
12. ****
13. ****
14. ****
15. **INORDER LDR IS THE STANDARD**
16. **PREORDER OR POSTORDER ALSO BUT CHOICE IS INORDER**
17. ****
18. **SKEWED TREE HAS ONLY ONE CHILD AND TC WOULD BE O(N)**
19. ****
20. **TC WISE EVERY ORDER IS SAME BUT FOR PERFORMANCE WISE, OPTIMISATION WISE INORDER IS BEST**
21. ****
22. ****
23. ****
24. **2 has returned TO 6 AFTER COMPLETING THE POSTORDER AND PREORDER**
25. **After PRINT WE BACK TO THE PREVIOUS ROOT NODE**
26. **BOTTOM TO TOP**
27. ****
28. **NUMBER OF NODES IN THE LONGEST ROOT TO LEAF PATH**
29. ****
30. **POST ORDER TRAVERSAL**
31. ****
32. ****
33. ****
34. ****
35. ** IS 15**
36. **7 ALSO DON’T HAVE CHILDREN WHICH RETURN 0 LIKE FOR ROOT.LEFT IS 6 THEN 6.LEFT RETURN 0 LH IS COMPLETE AND THEN RH 6.RIGHT WHICH 0 RETURN 1+0 IS 1 WHICH GO BACK TO ROOT 6 IS 1 AND THEN 11.LEFT COMPLETE THEN 11.RIGHT, 15 WHICH LH 7 AND THEN 7.LEFT IS 0 AND THEN GO BACK UP FROM NULL AND THEN 7. RIGHT IS 0 GO BACK TO UP TO 7 WHICH RETURN 1+0 IS 1 GO BACK TO 15 WHICH 15. LEFT GET IS 1 NOW 15.RIGHT IS 21**
37. **NOW 21.LEFT IS 18 AND THEN 18. LEFT IS 16 AND THEN 16.LEFT IS NULL RETURN 0 BACK UP TO 16 AND THEN 16 .RIGHT IS NULL RETURN 0 GO BACK TO 16 WHICH RETURN 1 GO BACK TO 18.LEFT IS DONE WITH 1 NOW 18.RIGHT WHICH US NULL RETURN 0 GO BACK TO UP NOW 1 + (18. LEFT,18.RIGHT) , 1+ 18.LEFT IS 16 WHICH RETURN 1 NOW 2 GO BACK TO 21 WHICH 21.LEFT WHICH 18 RETURN 2 NOW GO TO 21.RIGHT RETURN 0 RETURN 2 RETURN 1+ MAX (2,0) NOW GO BACK TO 15 RETURN 3**
38. **15.LEFT WHICH RETURN 1 FROM 7 AND 15.RIGHT RETURN 3 (1 +2) FROM 21 NOW 1 +(3,1) IS 4 RETURN BACK TO ROOT 11 WHICH 11.LEFT IS 1 AND 11.RIGHT IS 4 ANSWER IS 1+MAX(4,1)**
39. **IS 5**
40. ****
41. ****
42. **FOR ROOT 1 THE 2.LEFT IS 3 2.RIGHT IS4 MUST BE MIRROR IMAGE**
43. **2.LEFT IS 4 AND 2.RIGHT IS 3 MUST BE IDENTICAL IMAGE**
44. ****
45. **Tc is o(n)**
46. **Sc is o(n)**
47. ****
48. ****
49. ****
50. ****
51. **10 LEFT IS LEFT FROM 10 AND RIGHT IS FROM RIGHT**
52. ****
53. ****
54. **5 is root node INORDER 5 LEFT IS LEFT SUBTREE AND 5 RIGHT IS RIGHT SUBTREE**
55. **IN PREORDER 5 K RIGHT MAI 2 IS LEFT SUBTREE AND USSE AGE K RIGHT SUBTREE**
56. ****
57. **IN INORDER 7 K LEFT MAI LEFT SUBTREE AND 7 K RIGHT MAI RIGHT SUBTREE VASAE BNATE JAO**
58. **PREORDER KA PEHLA ELEMENT HMESHA ROOT NODE HOGA USKO INORDER MAI SEARCH KRKE INORDER K LEFT MAI LEFT SUBTREE H OR INORDER MAI ROOT K RIGHT MAI RIGHT SUBTREE HAI**
59. ****
60. ****
61. ****
62. ****
63. ****
64. ****
65. ****
66. ****
67. ****
68. ****
69. ****
70. **Tc=>o(n)**