A (3) (9)

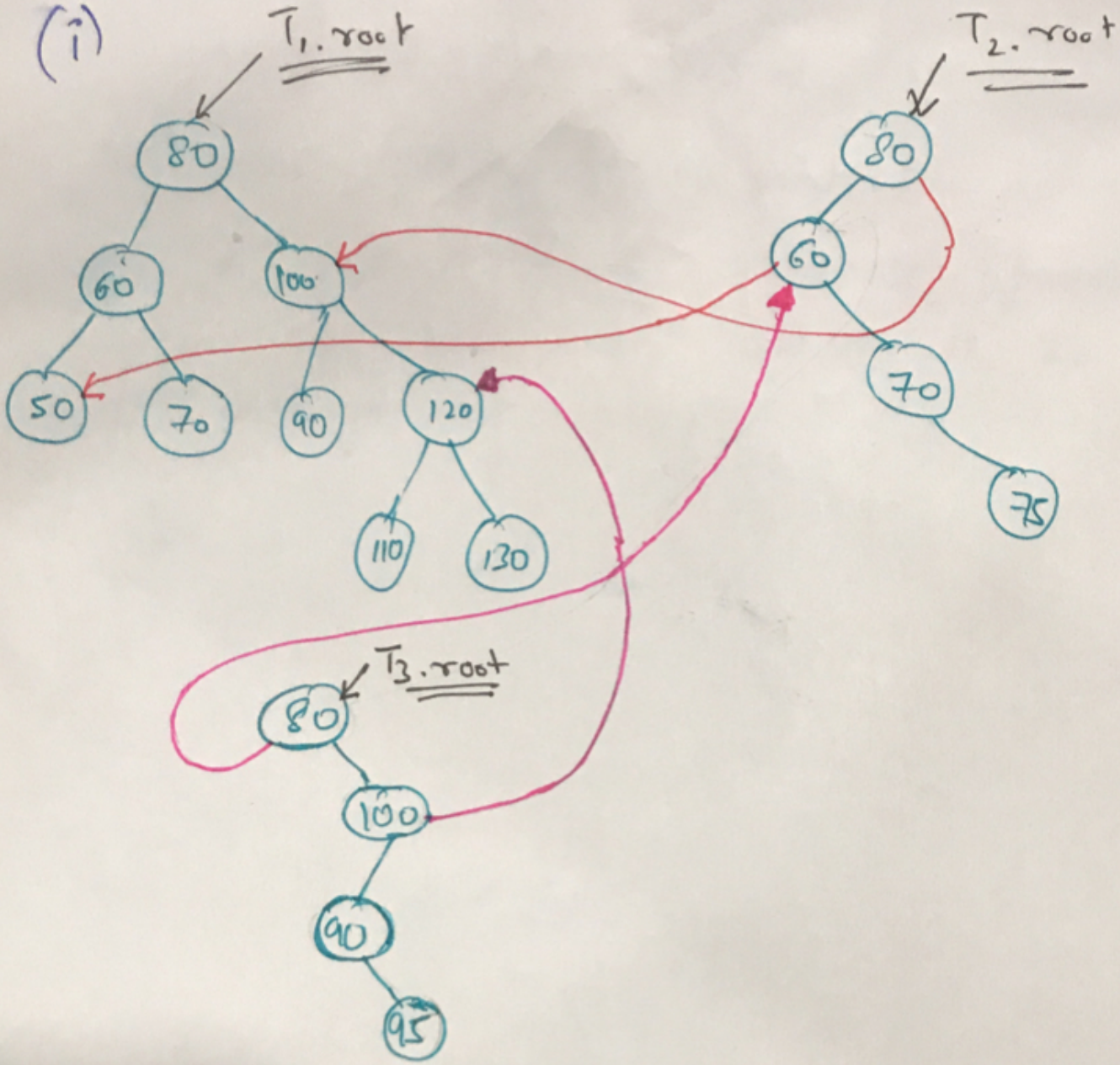
We always downwards from root, to do the insertion or deletion and, store that path from root to the node where insertion is to be done, which doesn't require any parent pointer.

→ We don't traverse the tree upwards.

→ We can have parent pointers, but it will be a wastage of memory.

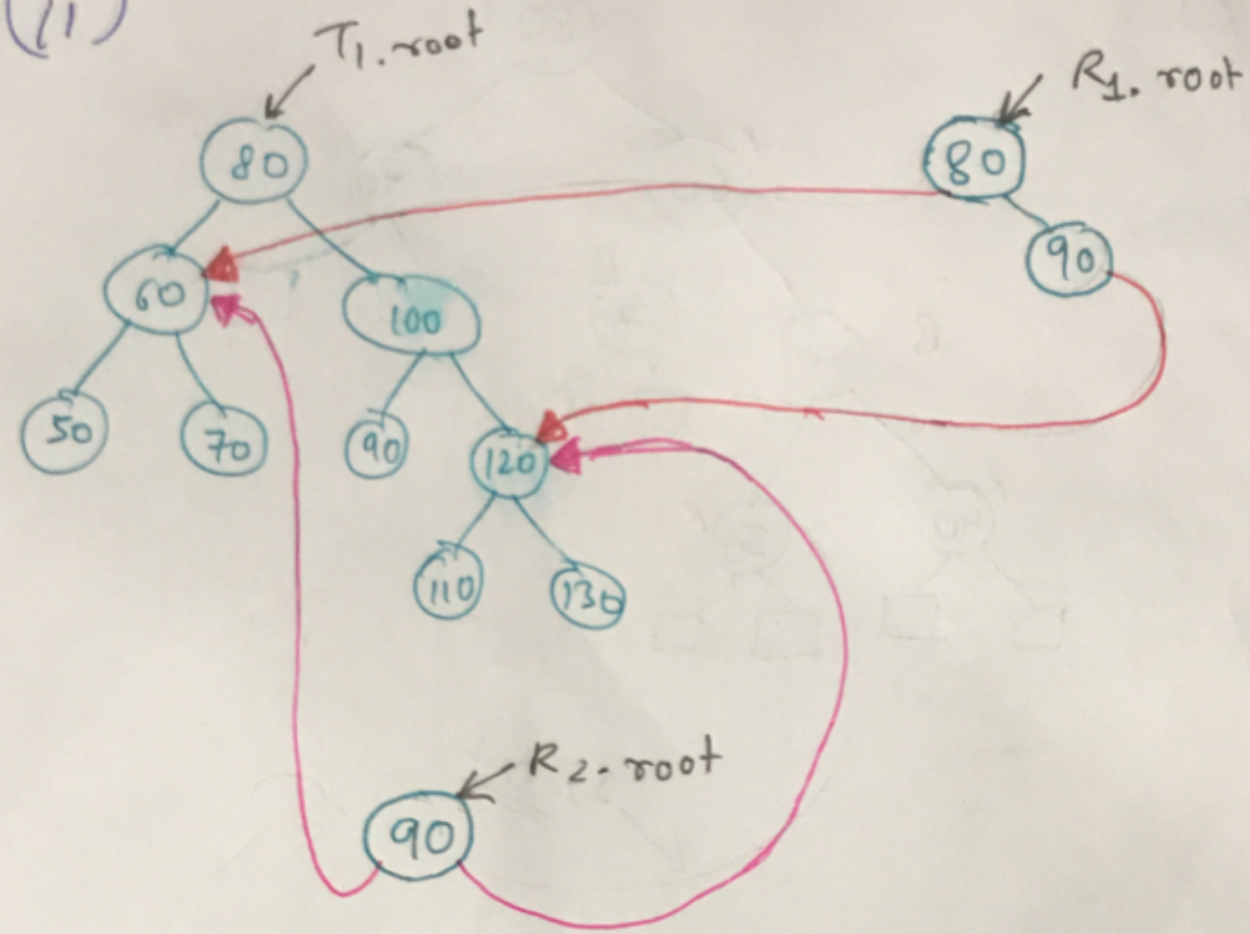
3b

(i)



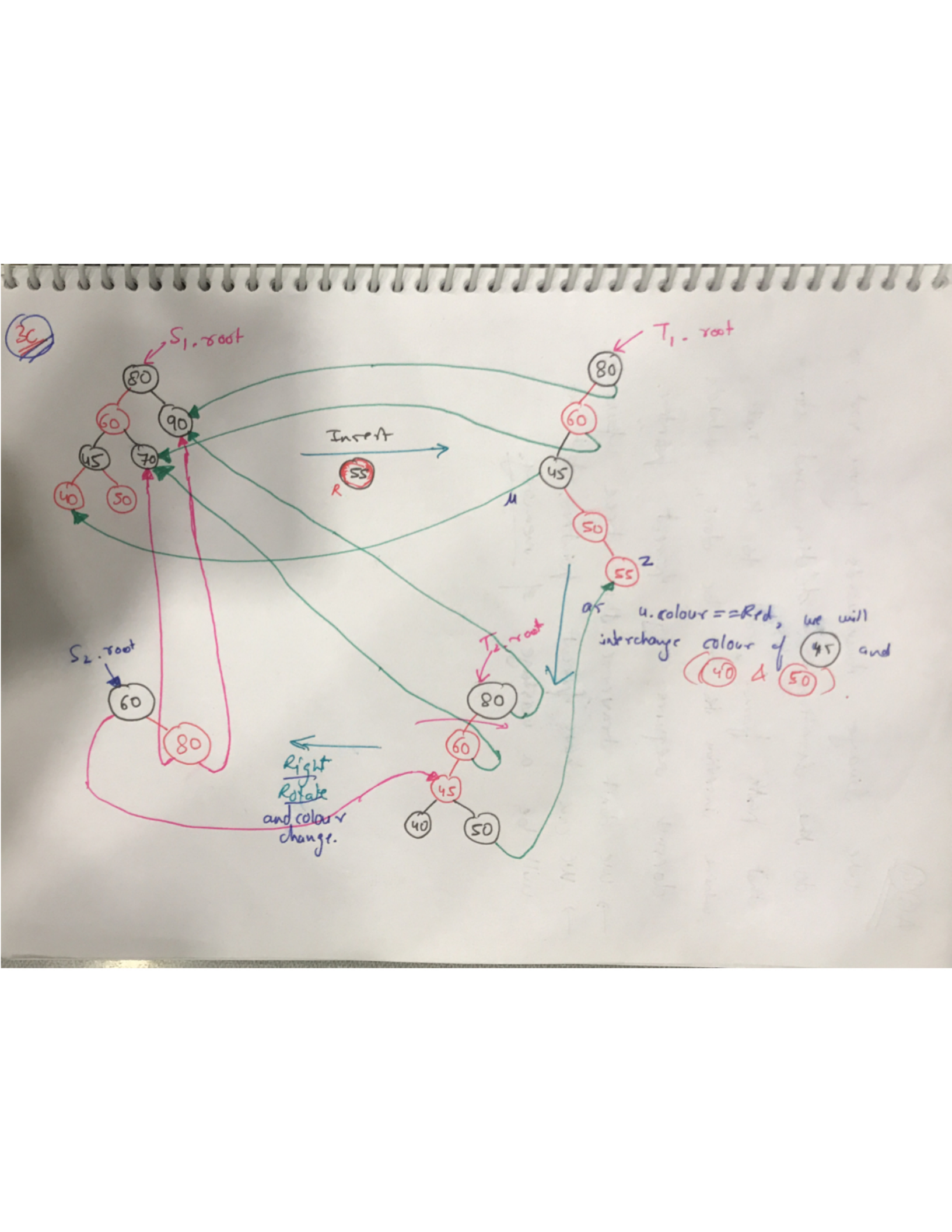
⊙ path upto 95 should be atleast made, and then we can link the leftover parts of the tree with T_1 and T_2.

3b
(ii)



(•) To create as few new nodes as possible, we are moving (90) in place of (100), this may only two new nodes have to created for R_1.

(•) after delting (80) from R_1, we have choosen (90) as the new node, as it will make sure, that we create minimum new nodes and links.


S_1.root
80
60
90
45
70
40
50
Insert
55
R
T_1.root
80
60
45
50
55
Z
u
ar
u.colour == Red, we will interchange colour of 45 and (40 & 50)
S_2.root
60
80
T_2.root
80
60
45
40
50
Right Rotate and colour change.