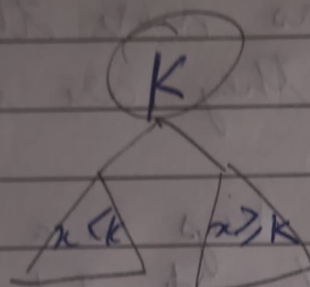
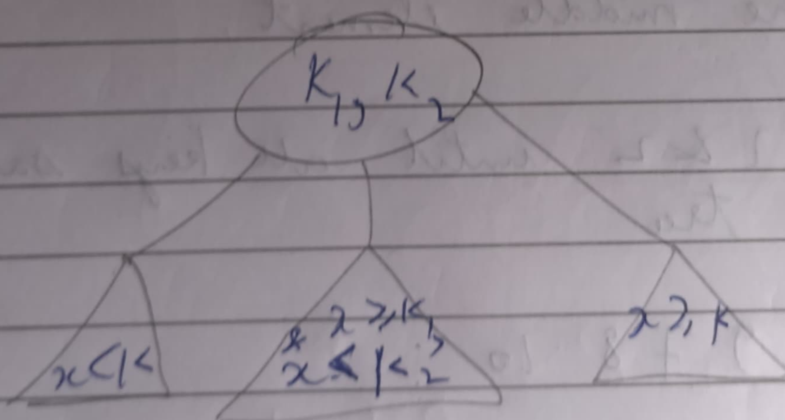
31) 2-3 Tree

(Example under representation change)
2-3 tree is a balanced BST with the following properties

1) It can have a 2-node structure

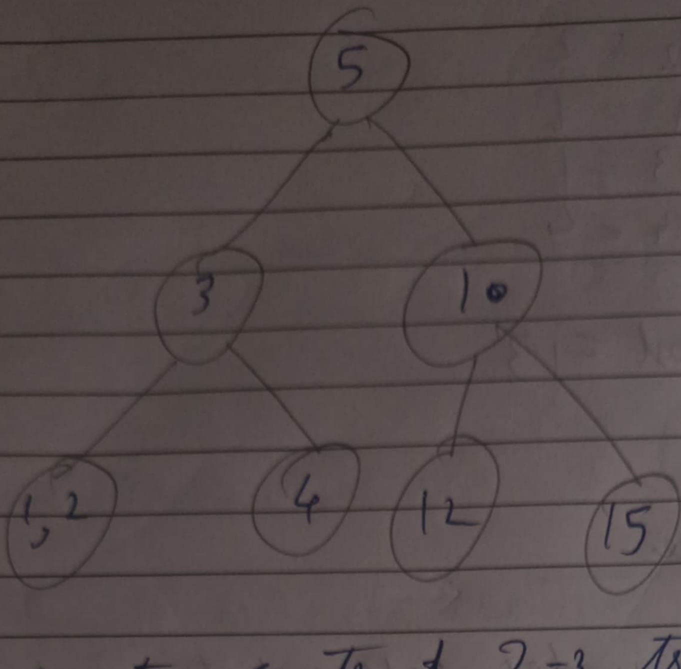


2) It can have a 3-node structure



3) All leaf nodes in a 2-3 Tree should be in the same level

Eg:-



Procedure to construct 2-3 tree

1) Insert the key in to a leaf after going through an appropriate branch

2) If a node has more than 1 or 2 keys make sure that they are in order.

~~3)~~ If there are 3 keys in a leaf, promote the middle one as parent. If the node with 3 keys is a root, split it & create a new root with the middle element.

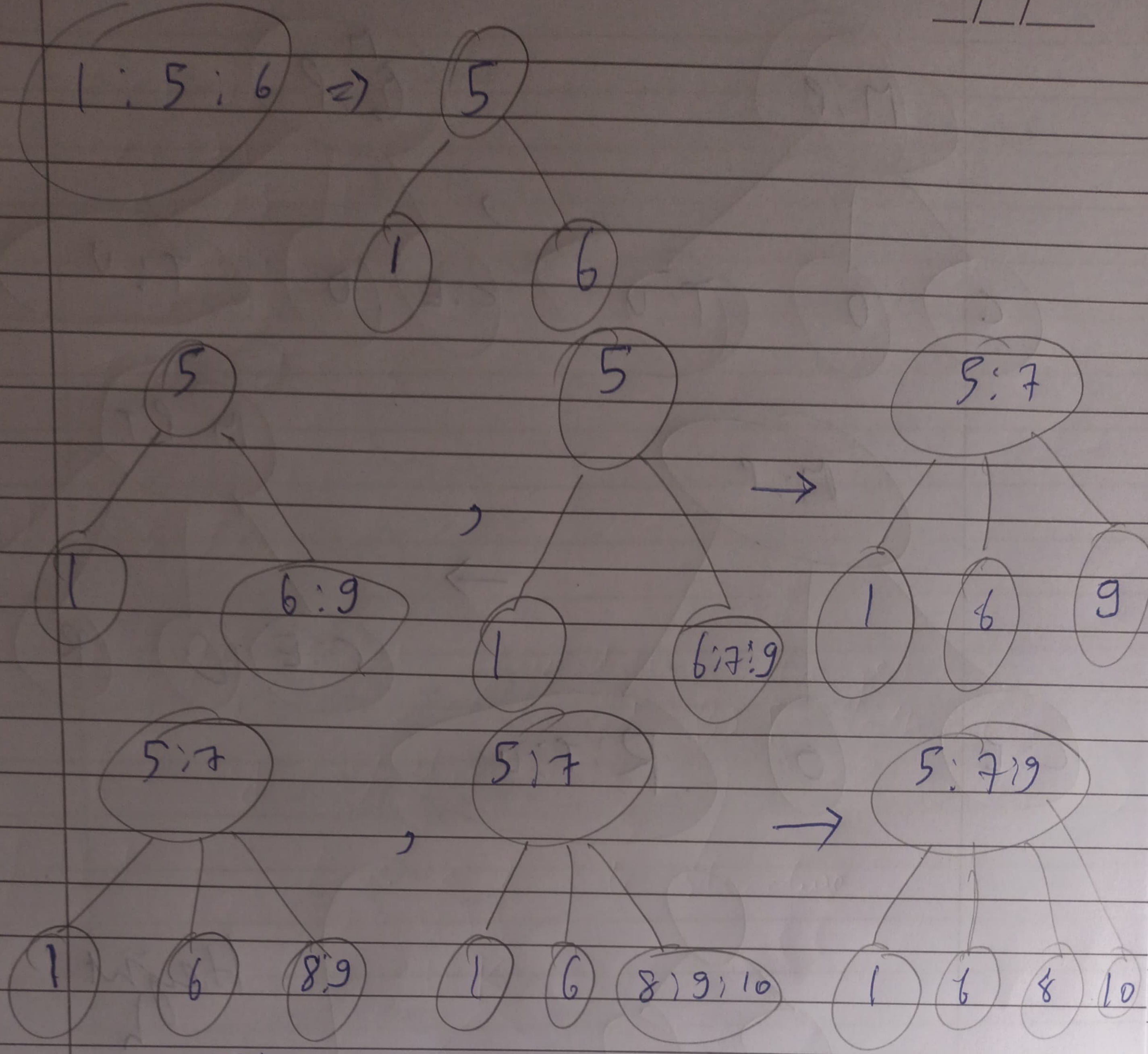
3) Repeat 1 & 2 until all keys satisfy properties of 2-3 tree

Eg:- 6 5 1 9 7 8 10

6

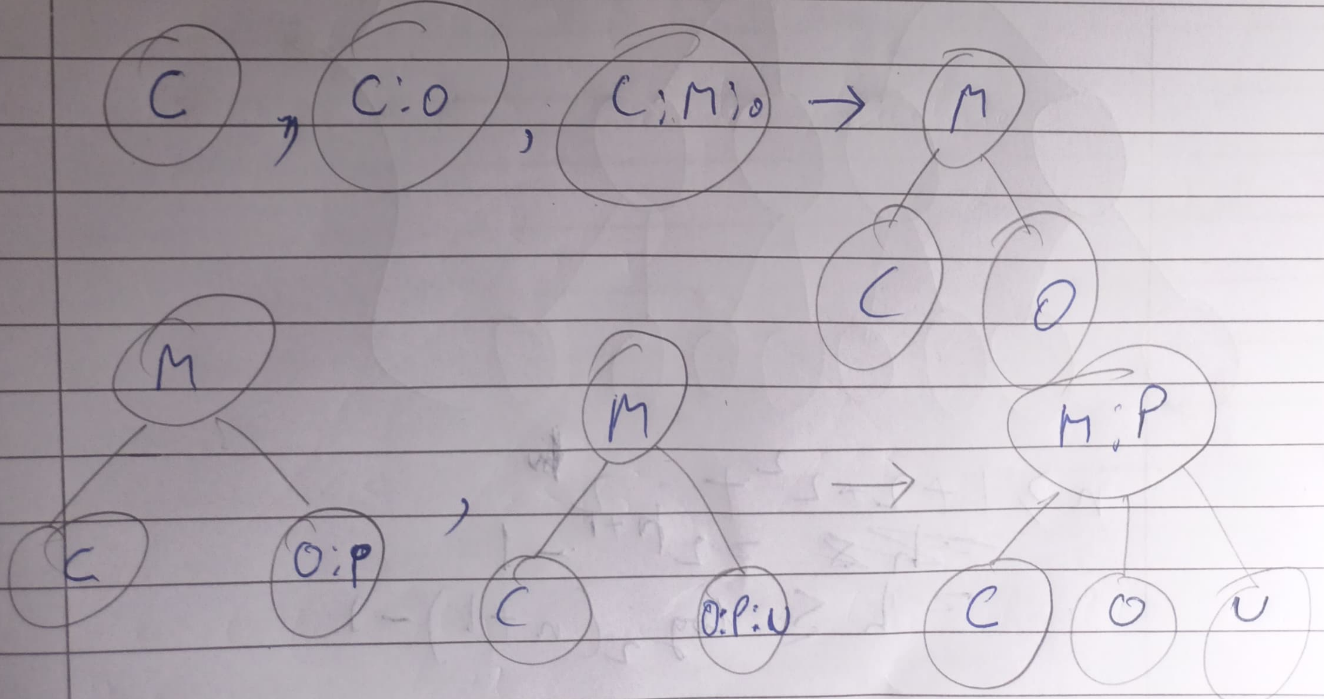
5;6

_/_/_

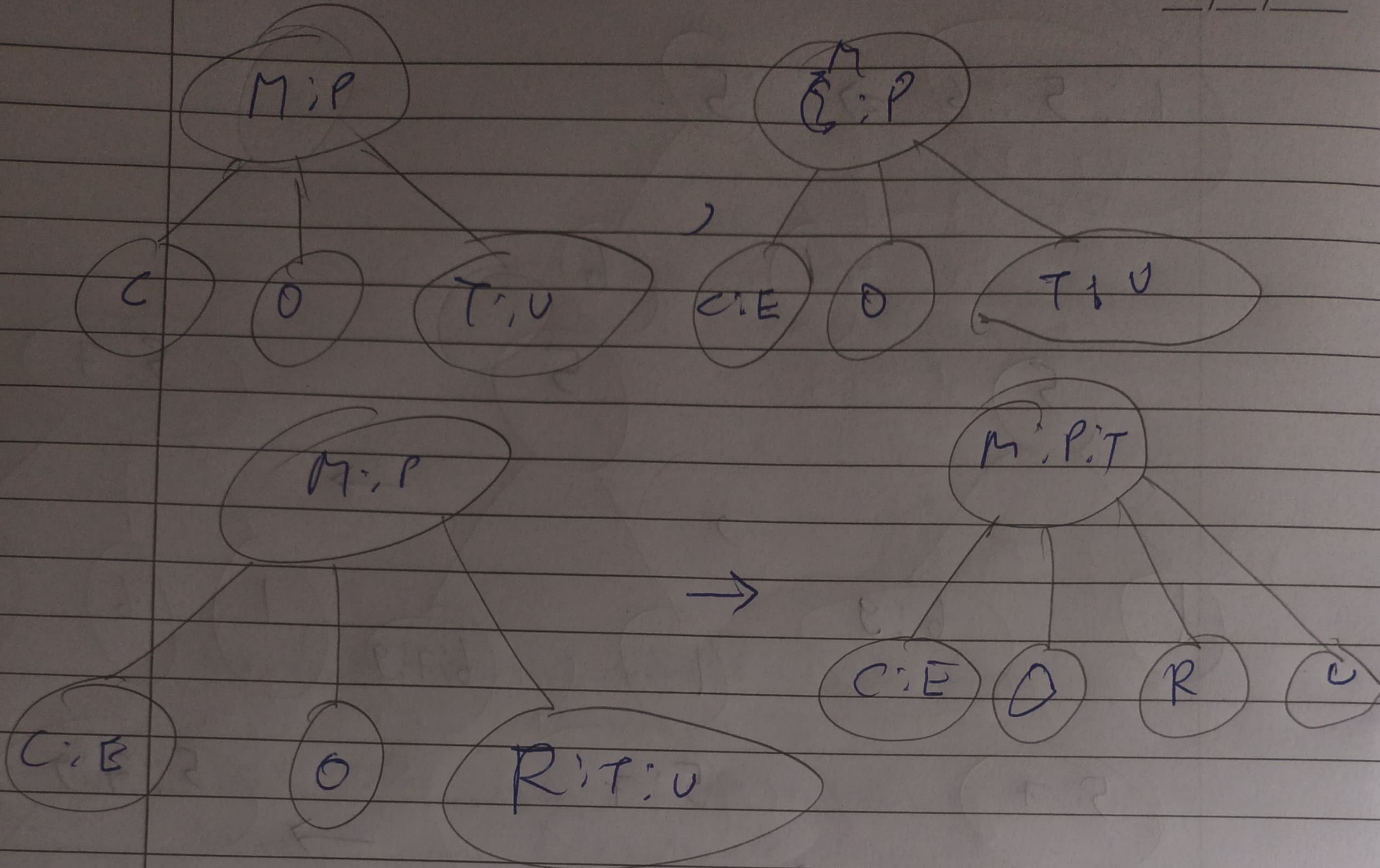


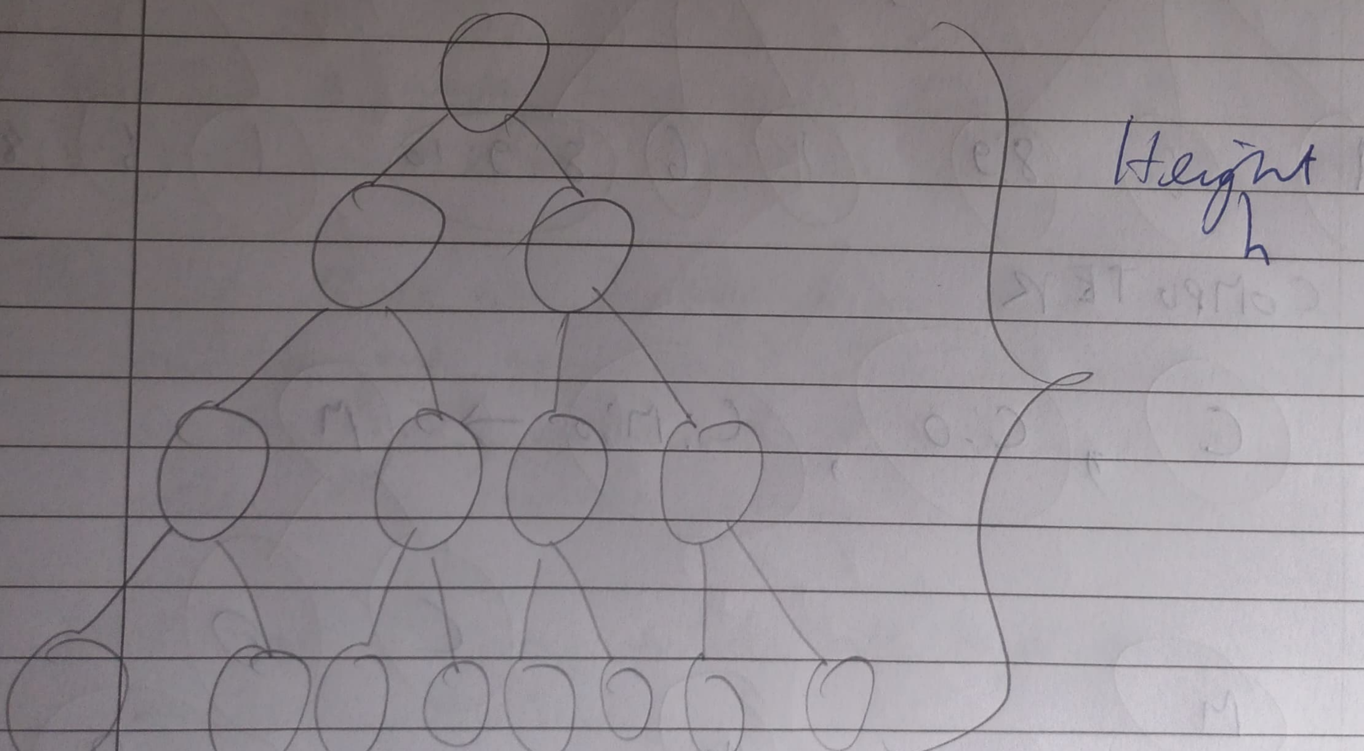
1:5:6
5
1
6
5
1
6:9
5
1
6:7:9
5:7
1
6
9
5:7
1
6
8:9
5:7
1
6
8:9:10
5:7:9
1
6
8
10

COMPUTER



C
C:O
C:M:O
M
C
O
M
C
O:P
M
C
O:P:U
M:P
C
O
U



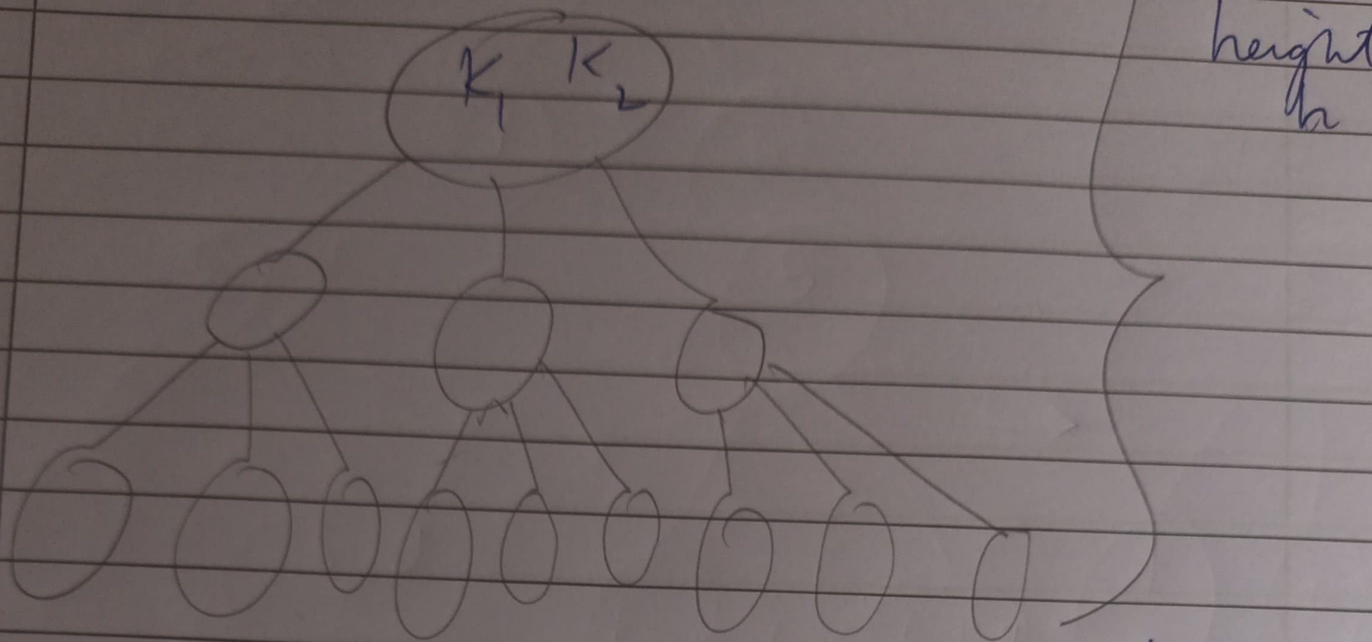


$$n \geq 1 + 2 + 2^2 + \dots 2^h$$

$$= 2^{h+1} - 1$$

$$\Rightarrow h \leq \log_2(n+1) - 1$$

3. node



$$n \leq 2 \cdot 1 + 2 \cdot 3 + 2 \cdot 3^2 \cdots\cdots 2 \cdot 3^h$$

$$= 2(1 + 3 + 3^2 + \cdots 3^h)$$

$$= 3^{h+1} - 1$$

$$h \geq \log_3(n+1) - 1$$

$$\therefore \boxed{\log_3(n+1) - 1 \leq h \leq \log_2(n+1) - 1}$$

Lower bound

Upper bound

Time complexity :- $\Theta(\log n)$