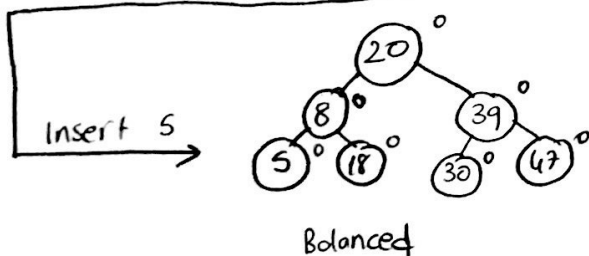
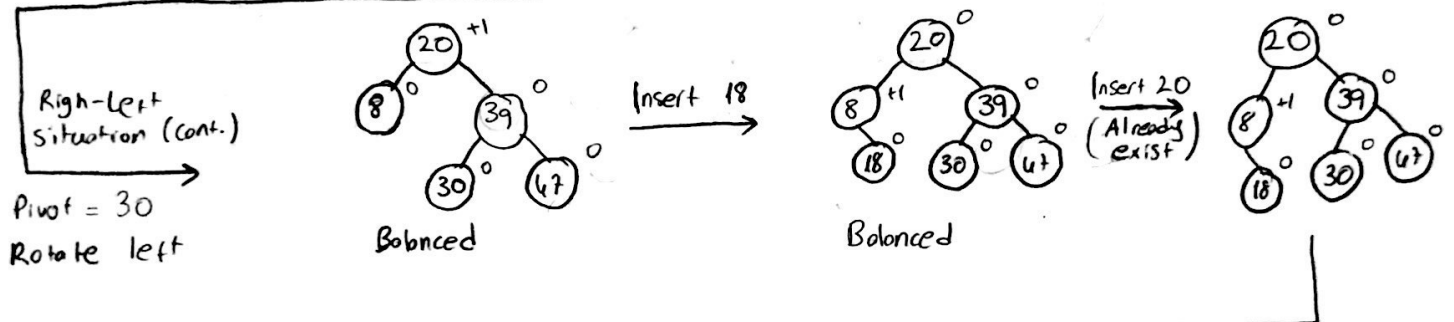
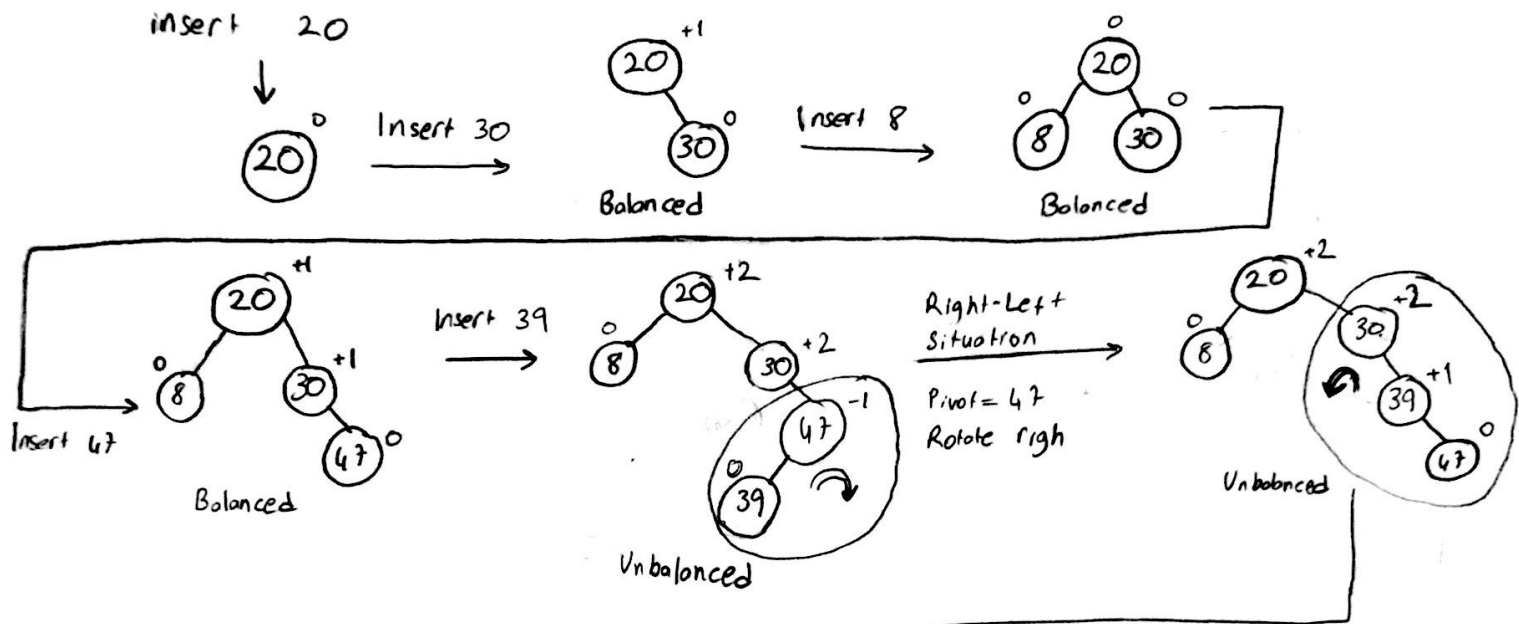
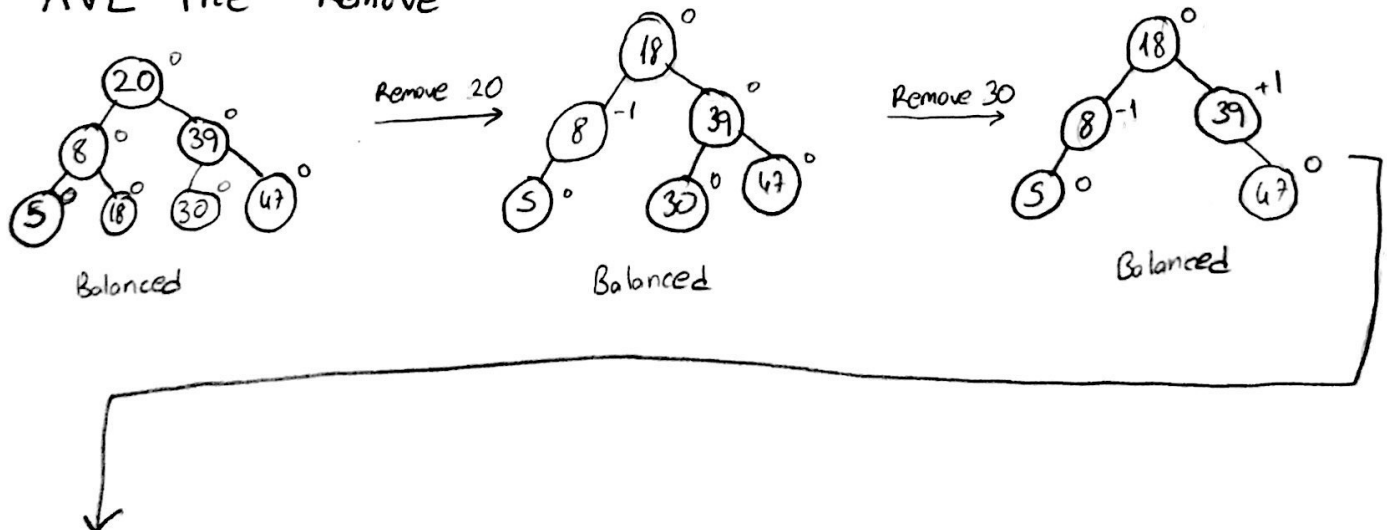
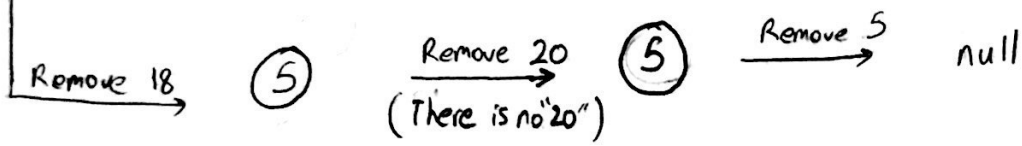
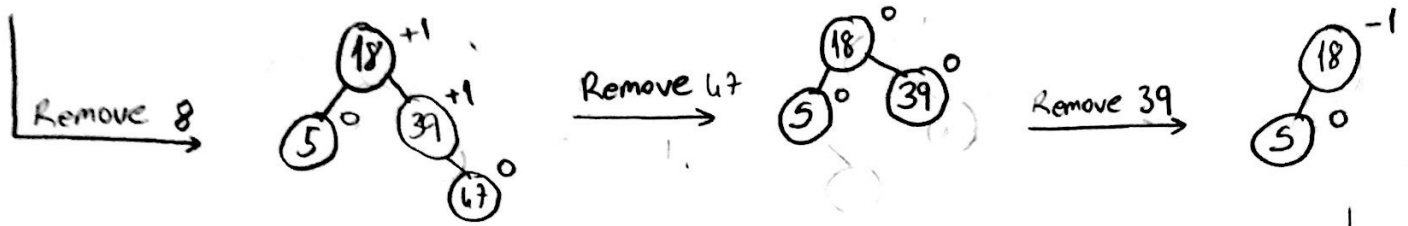


AVL Tree Insertion



AVL Tree Remove

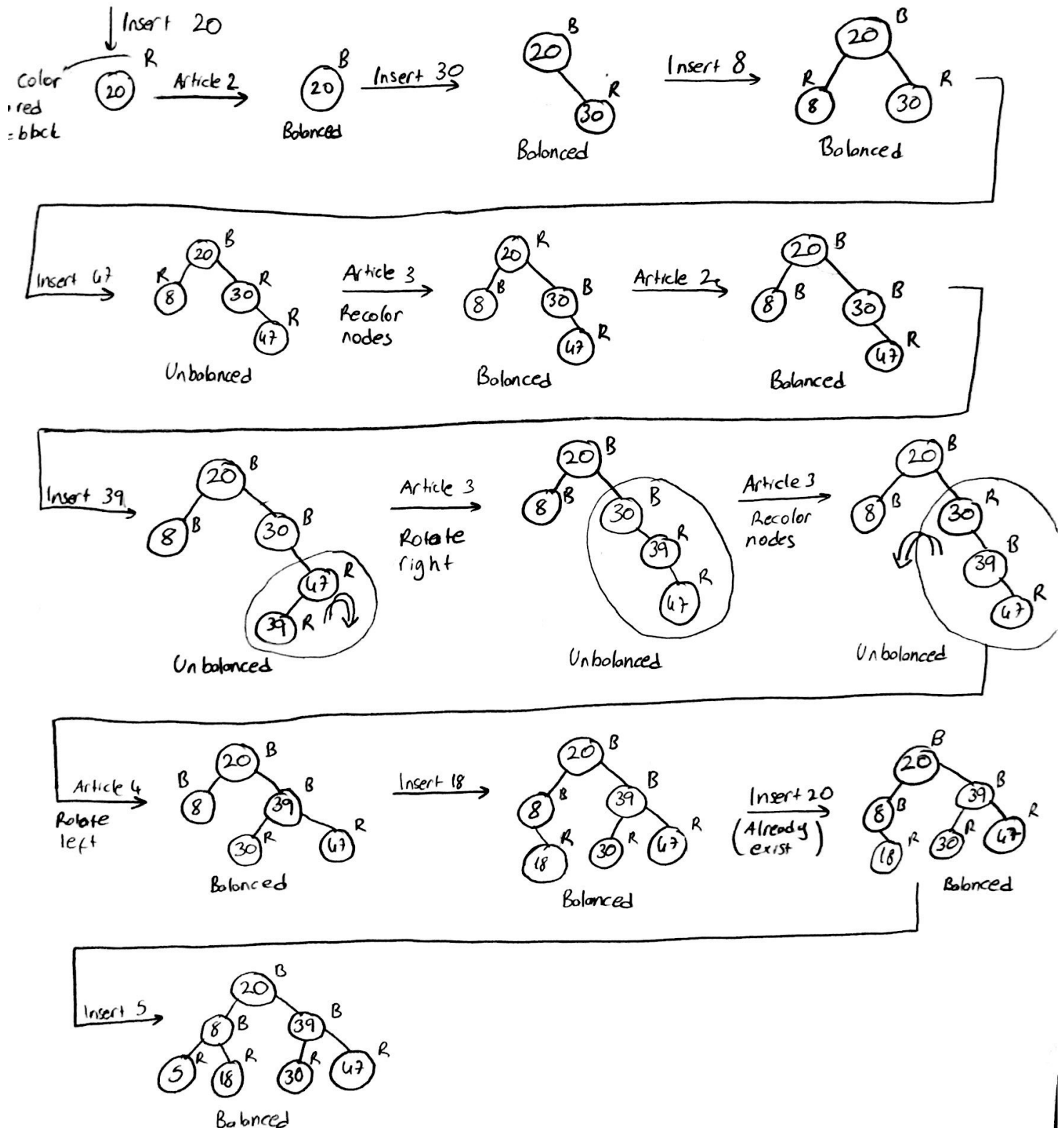




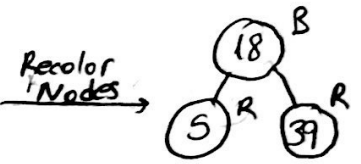
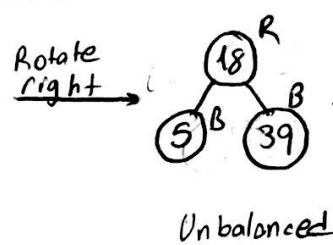
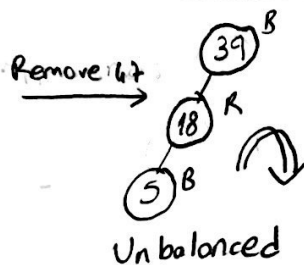
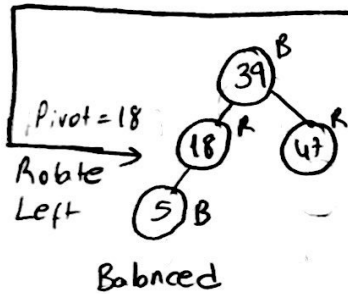
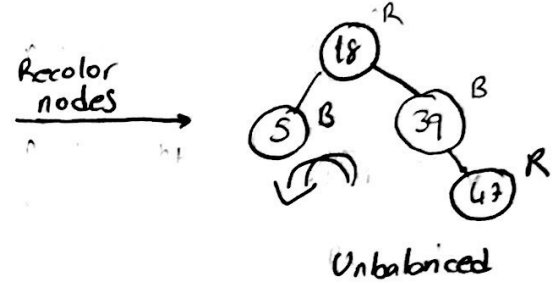
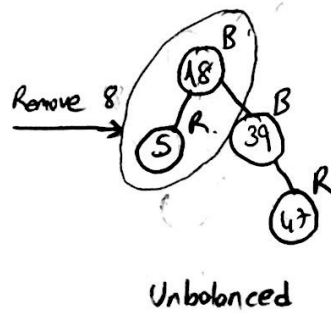
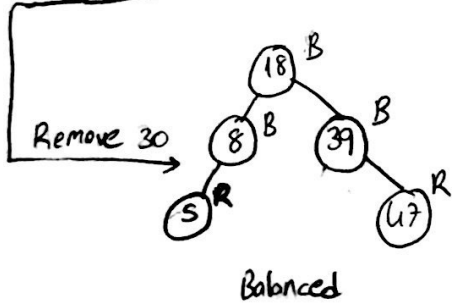
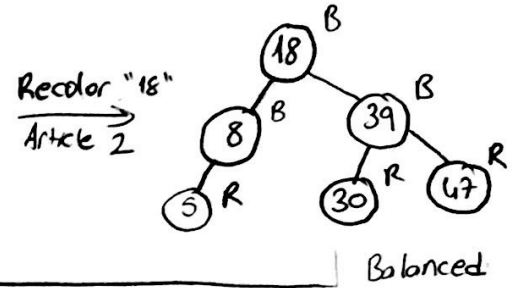
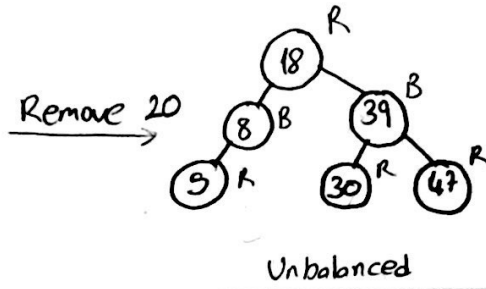
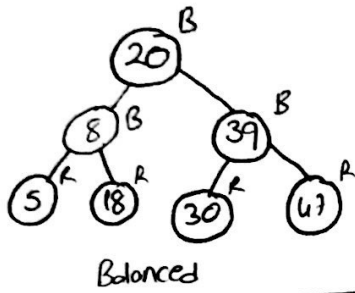
Red - Black Tree

- 1- A node is either red or black
- 2- The root is always black
- 3- A red node always has black children (a null reference is considered to refer to a black node)
- 4- The number of black nodes in any path from the root to a leaf is the same.

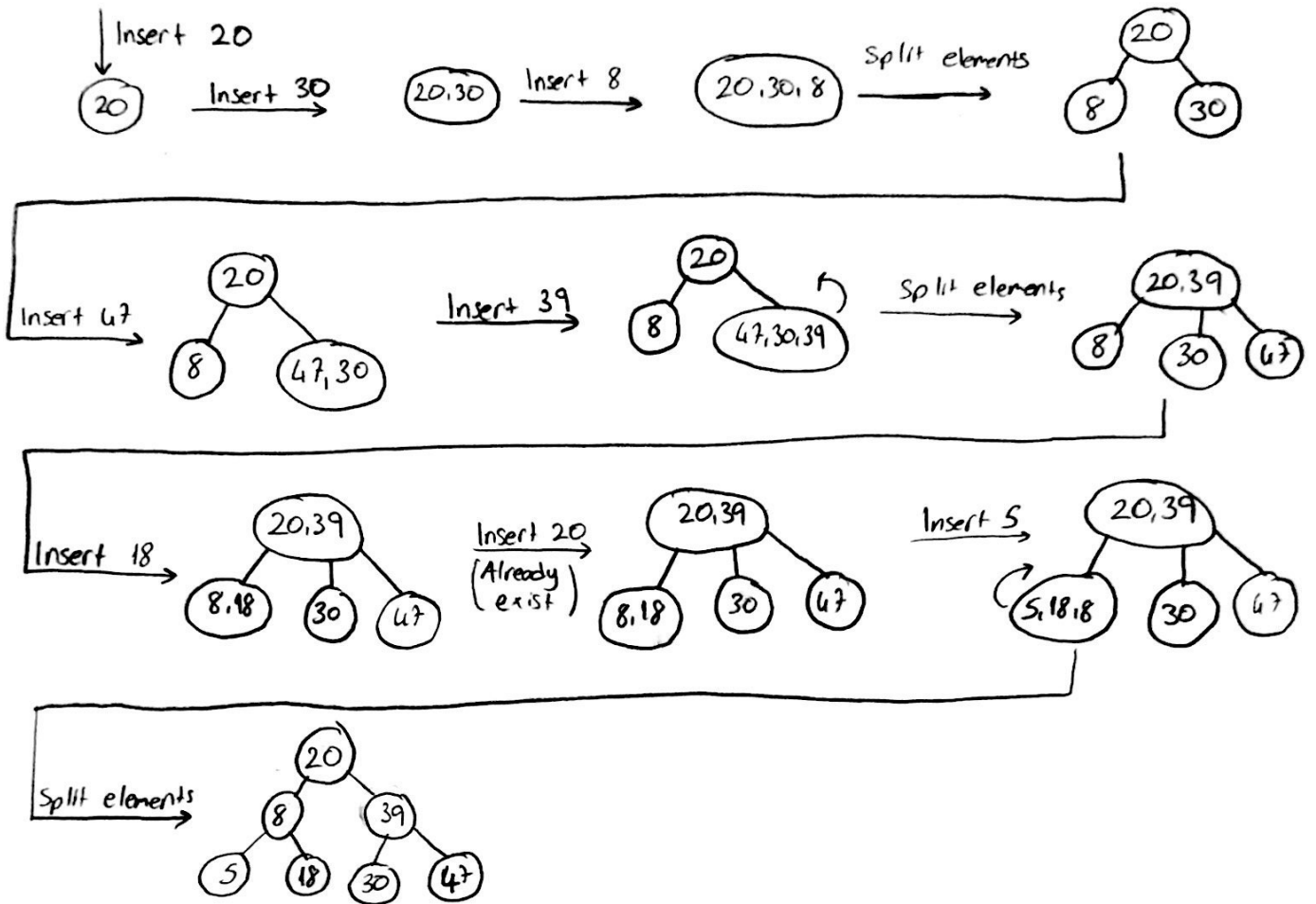
Red - Black Tree Insertion



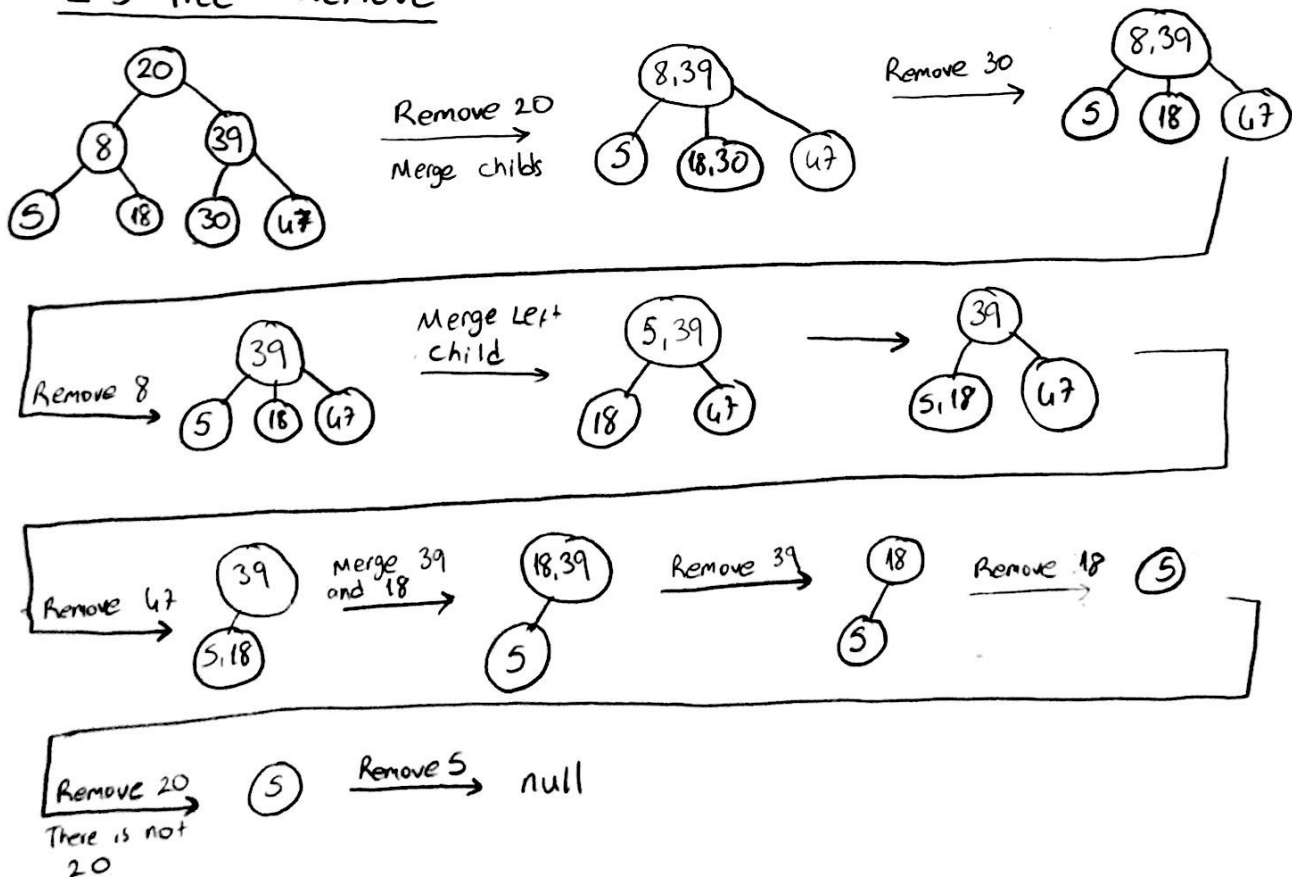
Red - Black Tree Remove



2-3 Tree Insertion



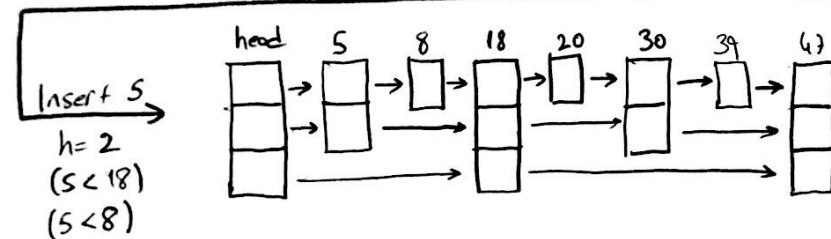
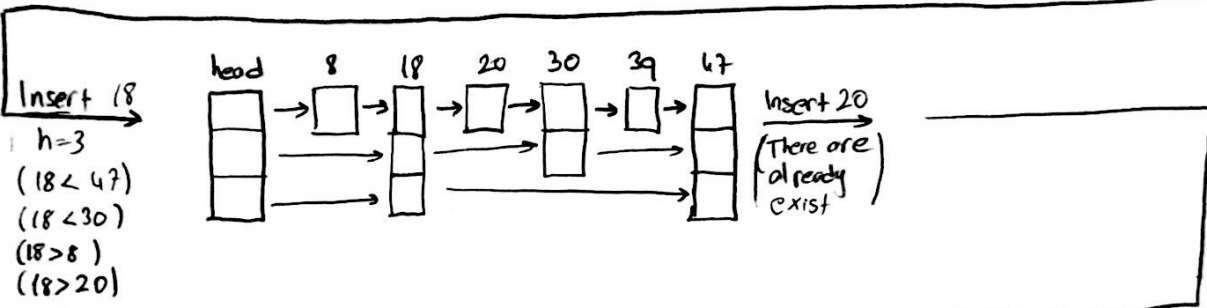
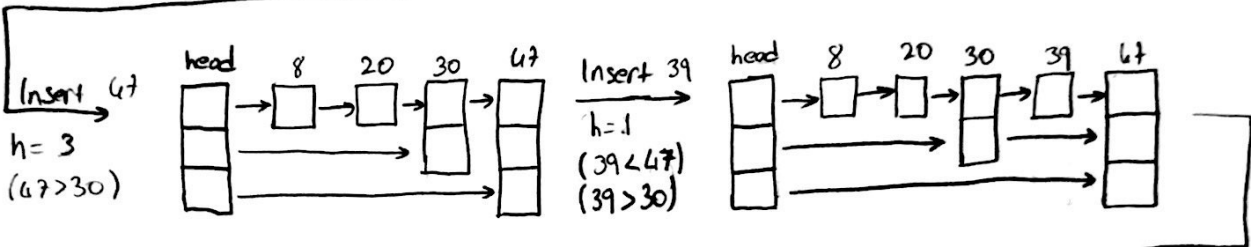
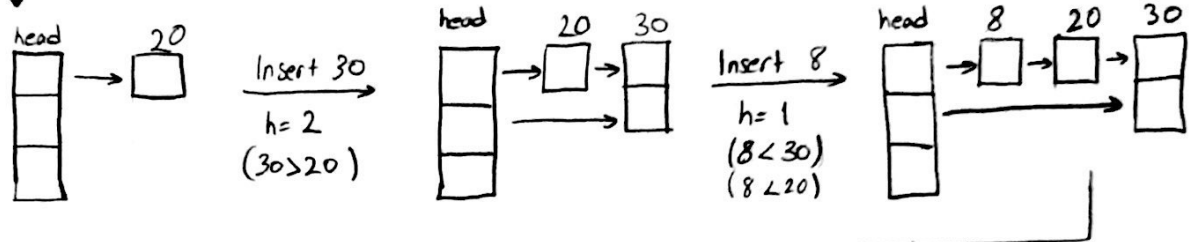
2-3 Tree Remove



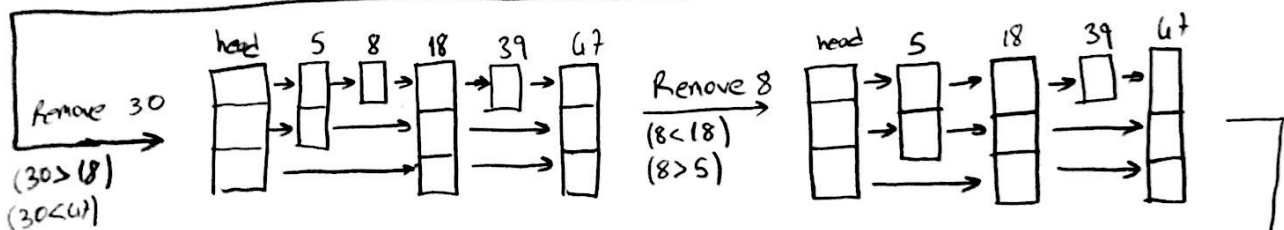
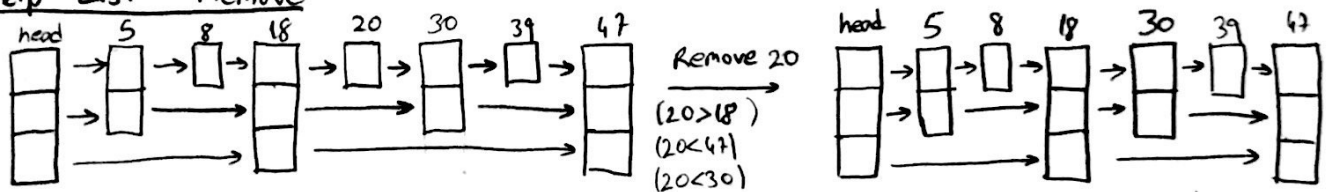
Skip List Insertion

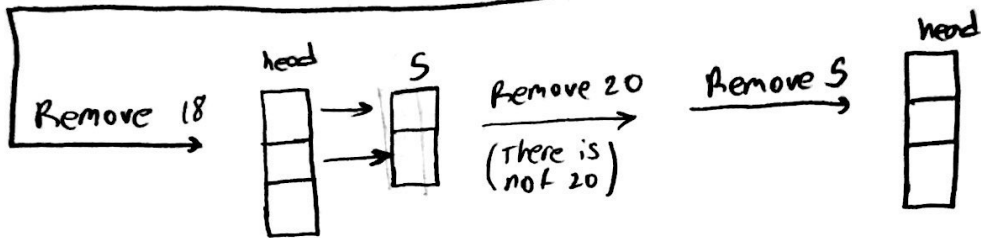
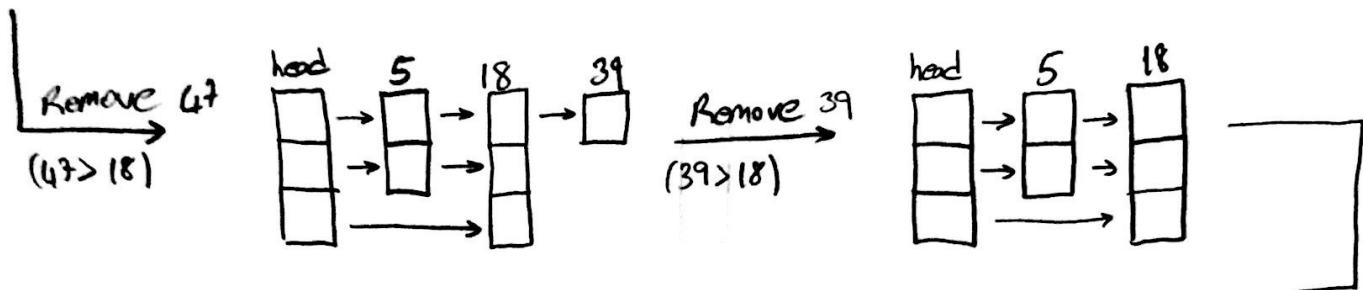
Highest node is 3 in this List

↓ Insert 20 ($h=1$)



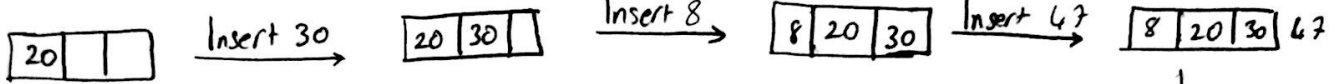
Skip List Remove





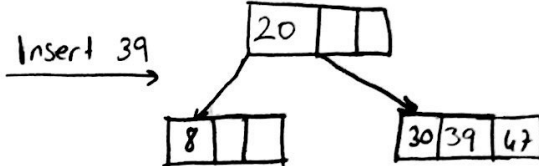
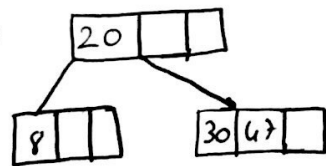
B-Tree with Order 4 Insertion

↓ Insert 20

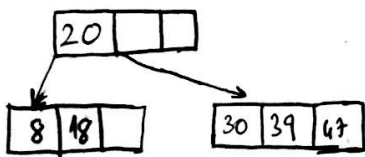


Because of node full splits node

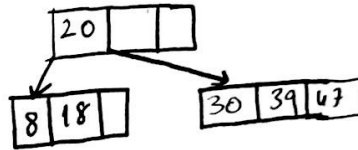
Left base design
Moves up 20



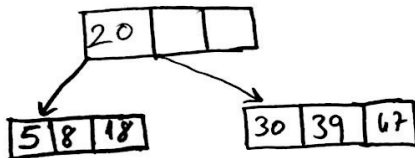
Insert 18



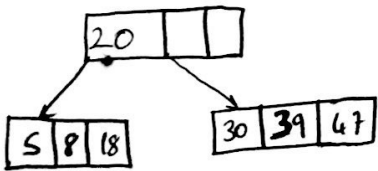
Insert 20
(Already exist)



Insert 5

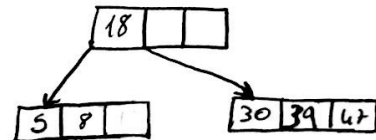


B-Tree with Order 4 Remove

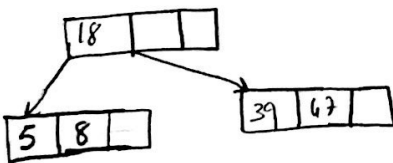


Remove 20

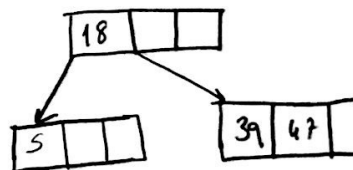
Left child has more than minimum number of keys. Replace 20 with 18



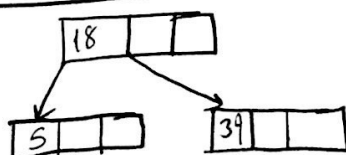
Remove 30



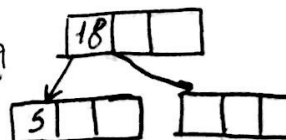
Remove 8



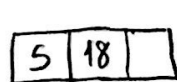
Remove 47



Remove 39



Because of the node empty, merge parent node



Remove 18



Remove 20

(There is not 20)



Remove 5

→ null