

**Q1**

Build the following data structures by using the sequence of integers,

20, 30, 8, 47, 39, 18, 40, 46

and remove all items one by one.

Esra Eryilmaz  
171044046

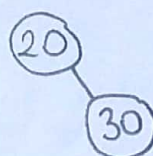
## - AVL Tree -

Insertion

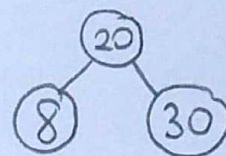
Insert 20 →



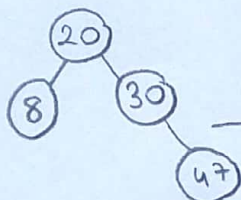
Insert 30 →



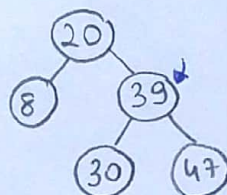
Insert 8 →



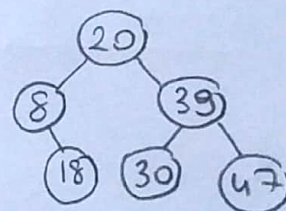
Insert 47 →



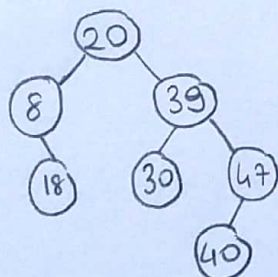
Insert 39 →



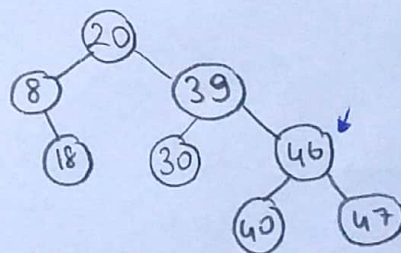
Insert 18 →



Insert 40 →



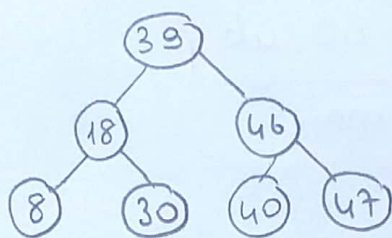
Insert 46 →



For maintaining balance, while some insertion I change some nodes locations. (while inserting 39 and 46)

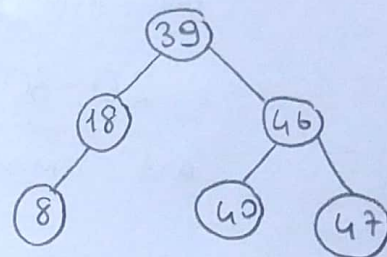
# Deletion

Remove 20 →

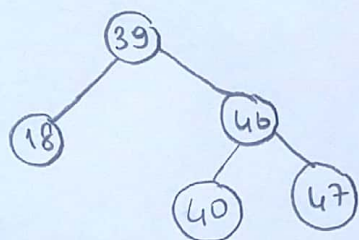


Left rotate (39)

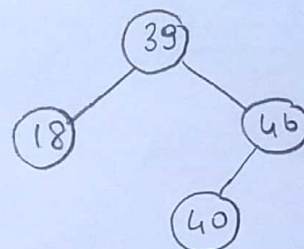
Remove 30 →



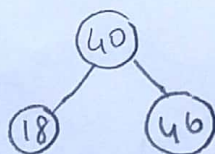
Remove 8 →



Remove 47 →

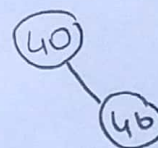


Remove 39 →



Left Rotate (40)

Remove 18 →



Remove 40 →



Remove 46 →



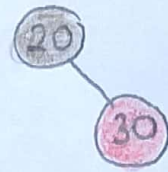
# - RED - BLACK Tree -

## Insertion

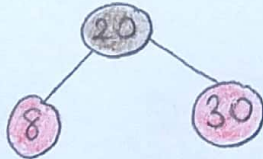
Insert 20 →



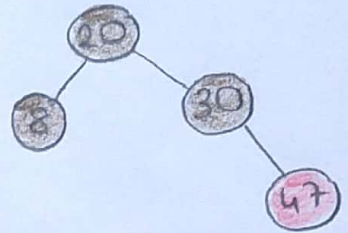
Insert 30 →



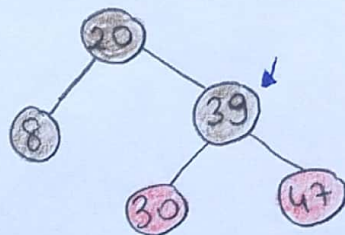
Insert 8 →



Insert 47 →

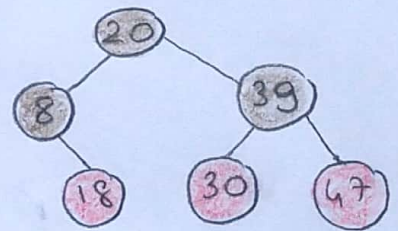


Insert 39 →

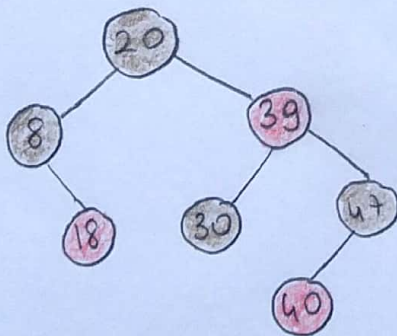


(Left Rotation)

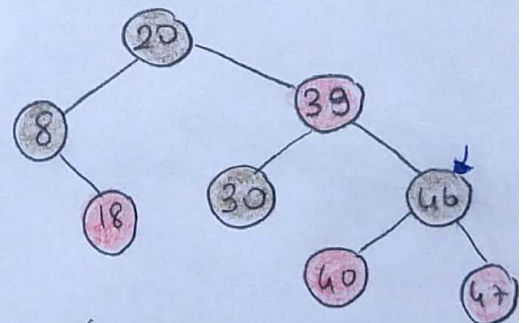
Insert 18 →



Insert 40 →



Insert 46 →



(Right Rotation)

\* A node is either red or black.

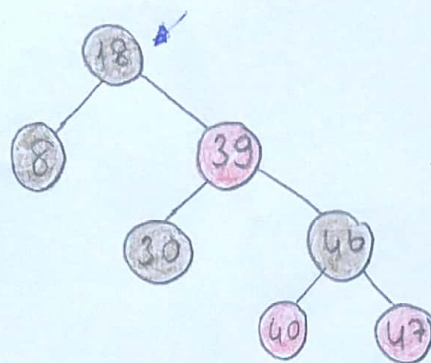
\* The root is always black.

\* A red node always has black children.

\* The number of black nodes in any path from the root to a leaf is the same.

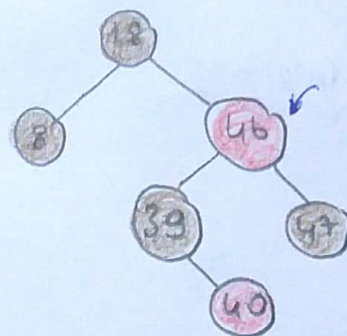
# Deletion

Remove 20



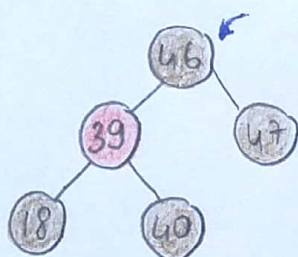
(18 becomes root)

Remove 30



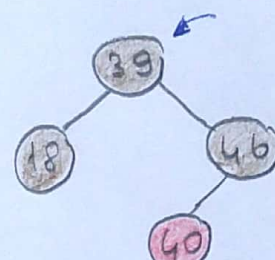
(Left Rotation)

Remove 8



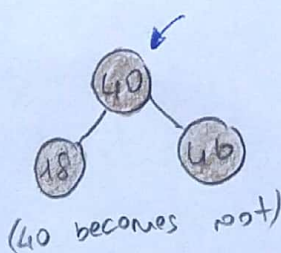
(Left rotation)

Remove 47



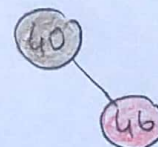
(Right Rotation)

Remove 39



(40 becomes root)

Remove 18



Remove 40



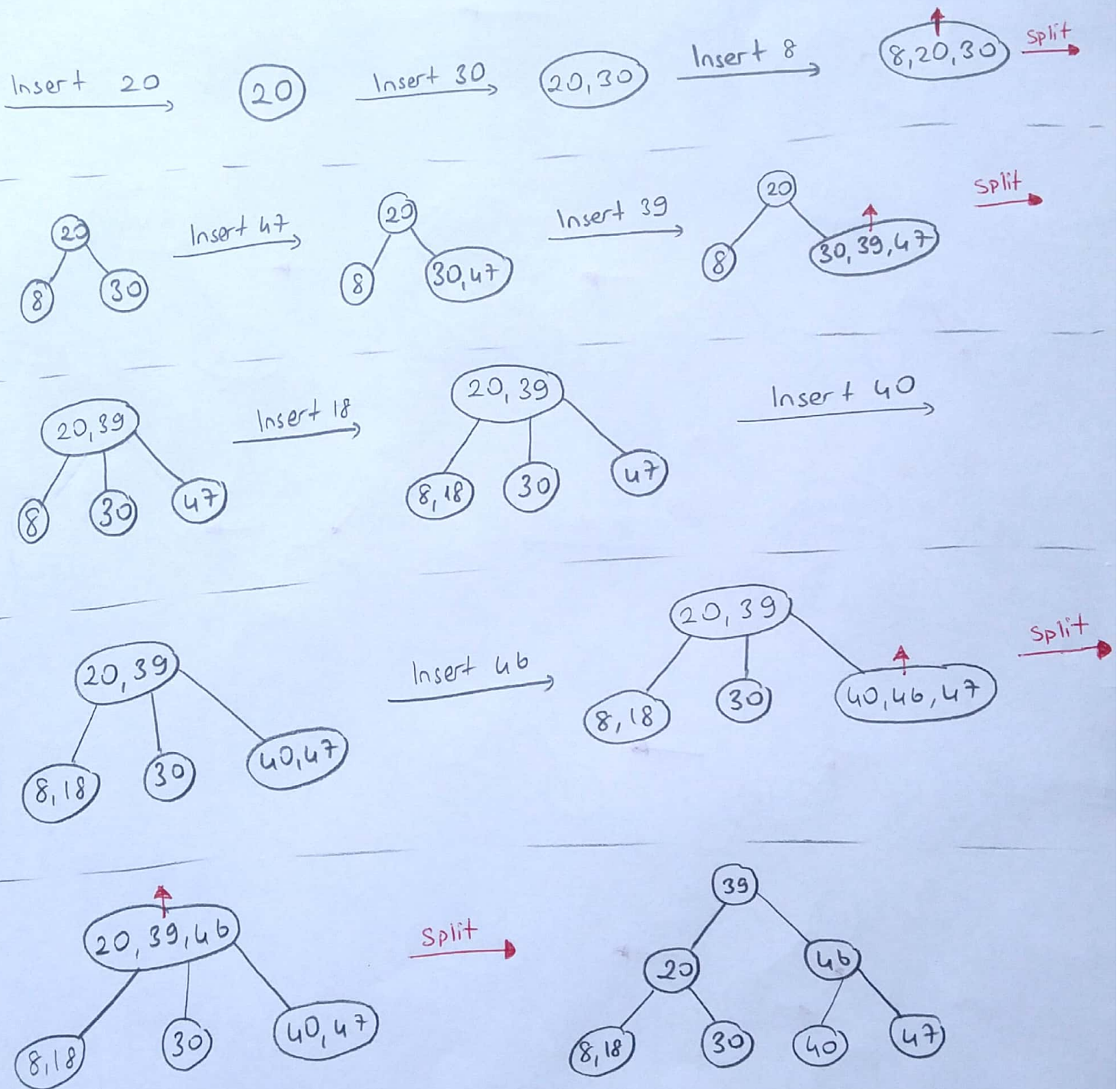
Remove 46

It is balanced but not as balanced as AVL tree



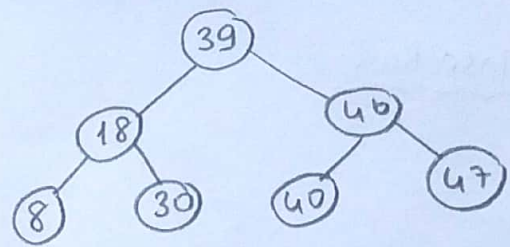
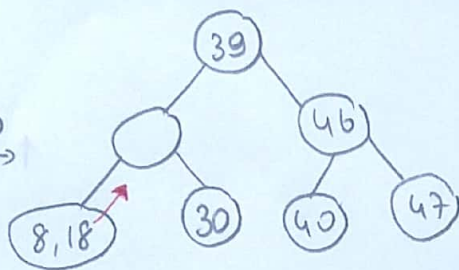
# 2-3 Tree

## Insertion

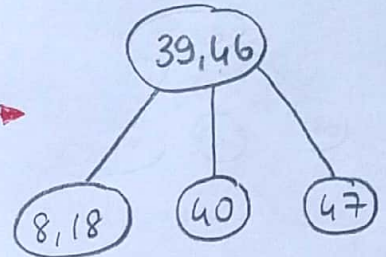
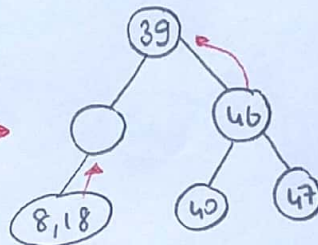
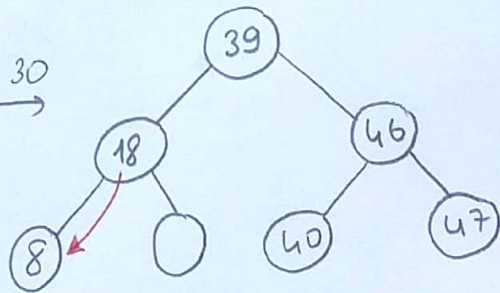


# Deletion

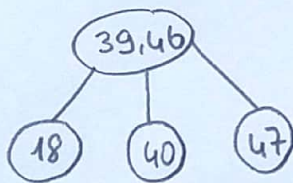
Remove 20



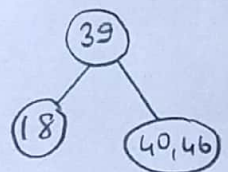
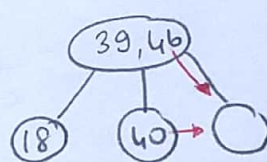
Remove 30



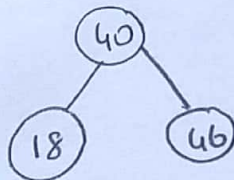
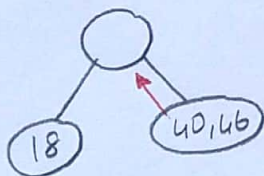
Remove 8



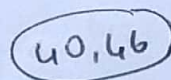
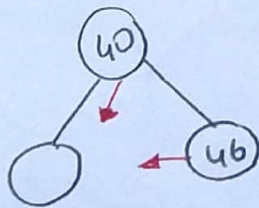
Remove 47



Remove 39



Remove 18



Remove 40



Remove 46

\* 2-3 Tree deletion same as B-Tree deletion.



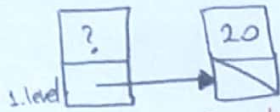
# Skip List

Max level  $\Rightarrow$

$$\log_2 n + 1 \rightarrow \log_2 8 + 1 = \underline{\underline{4}}$$

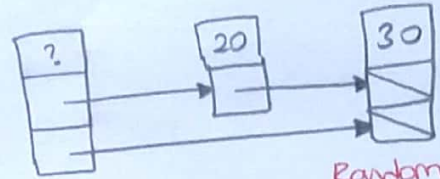
## Insertion

Insert 20



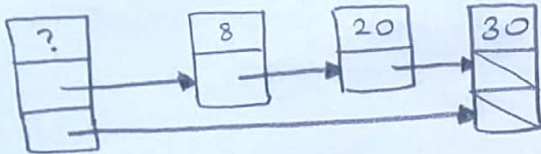
Random Num = 1

Insert 30



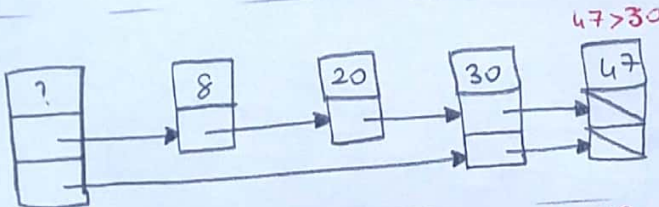
Random Num = 2

Insert 8



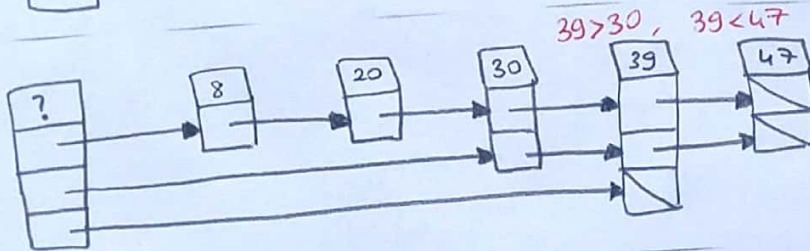
Random Num = 1 (up to 2)

Insert 47



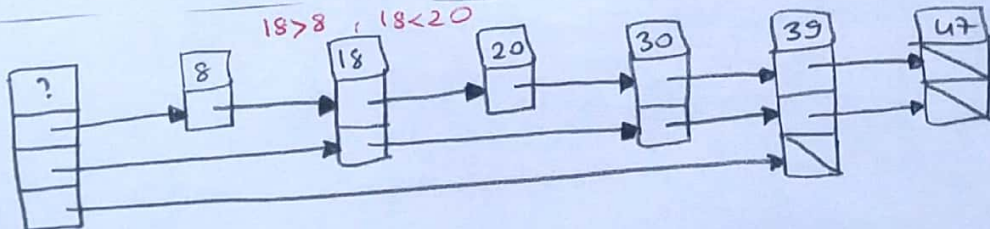
Random Num = 2 (up to 3)

Insert 39



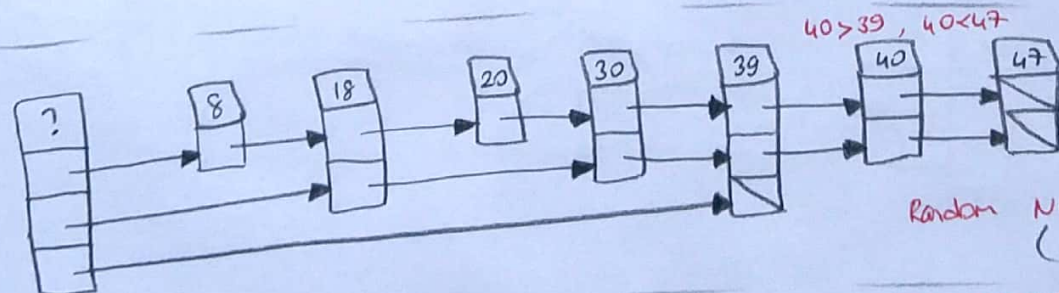
Random Num = 3  
(up to 3)

Insert 18



Random Num = 2  
(up to 3)

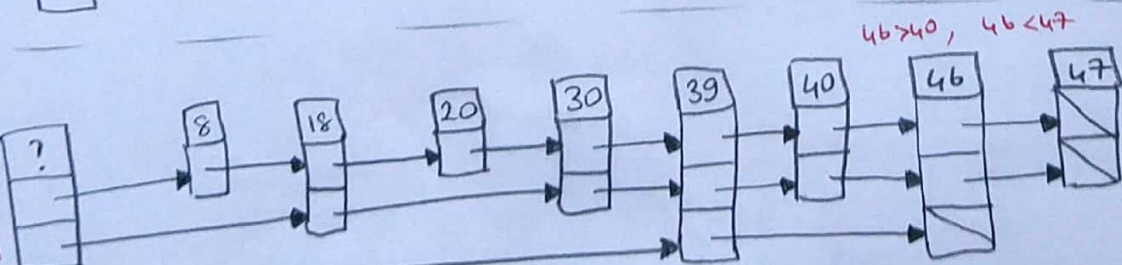
Insert 40



Random Num = 2  
(up to 3)

Insert 46

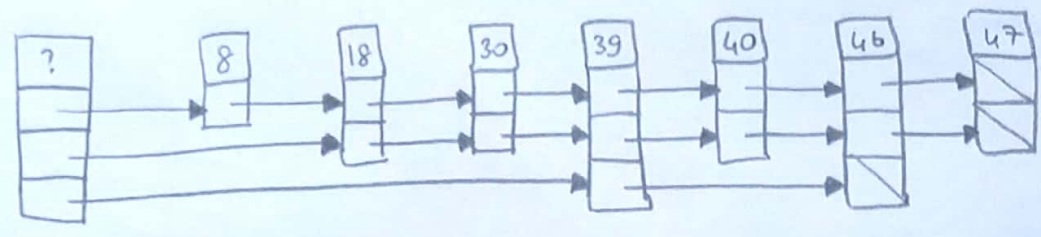
1.level  
2.level  
3.level



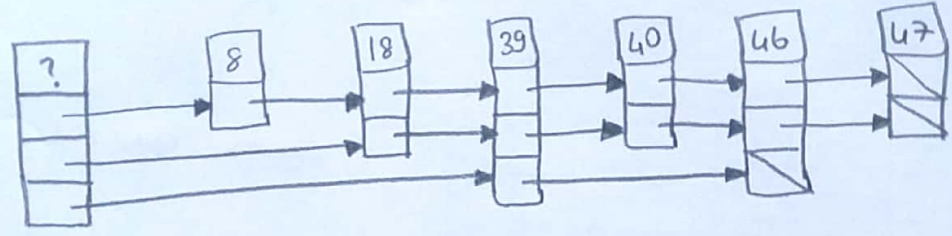
Random Num = 3  
(up to 4)

# Deletion

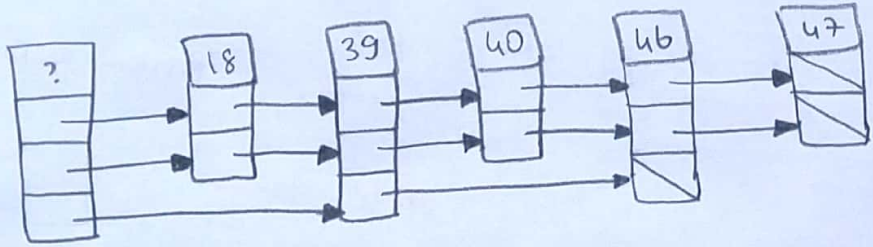
Remove 20



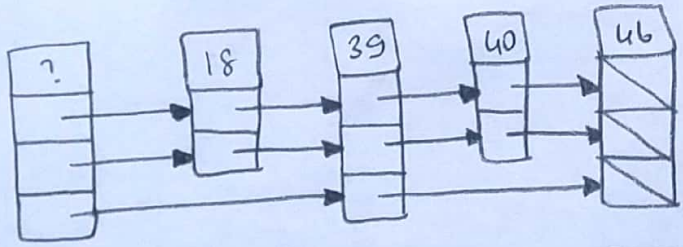
Remove 30



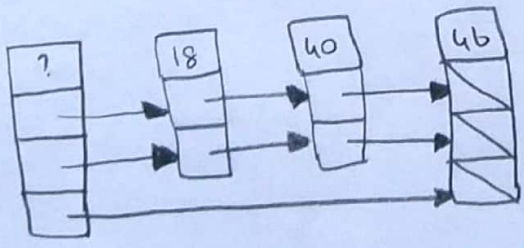
Remove 8



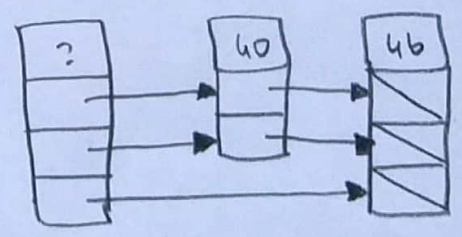
Remove 47



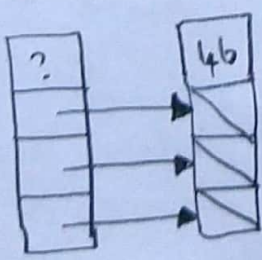
Remove 39



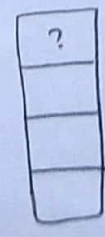
Remove 18



Remove 40



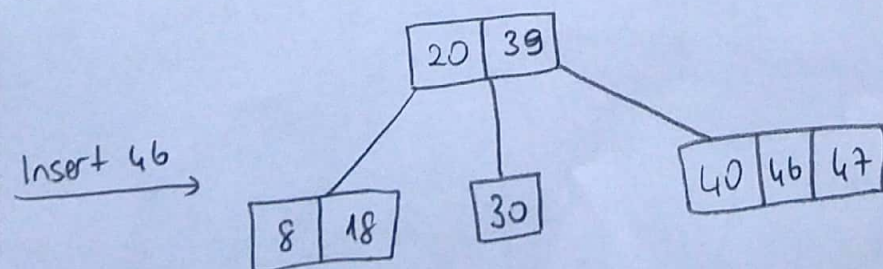
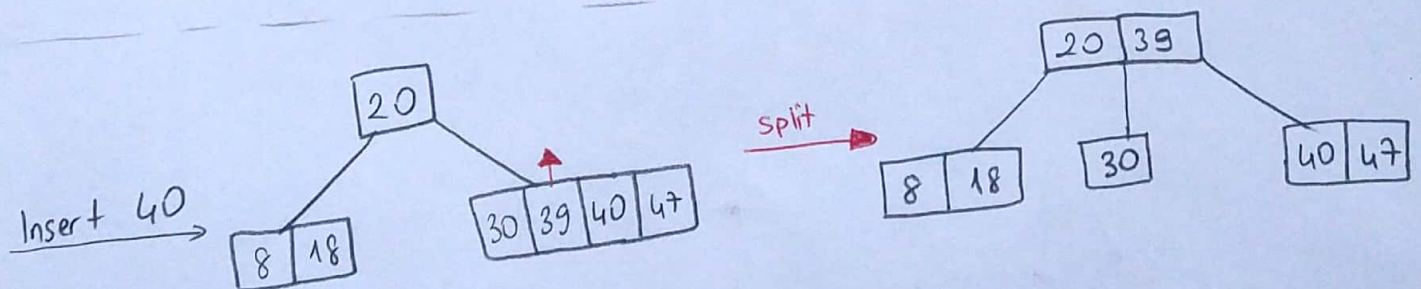
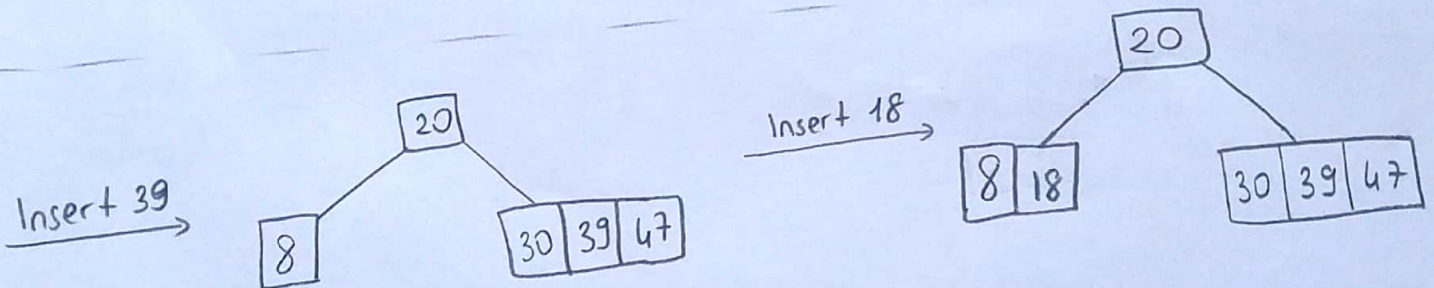
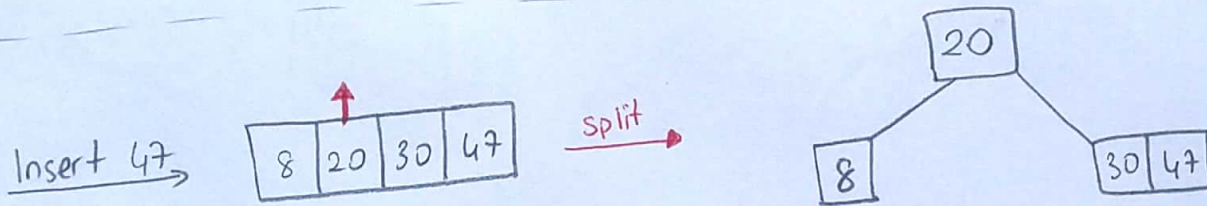
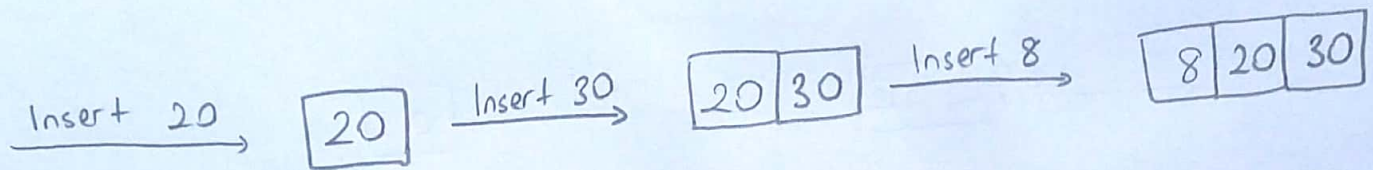
Remove 46





# B-Tree with order 4

## Insertion

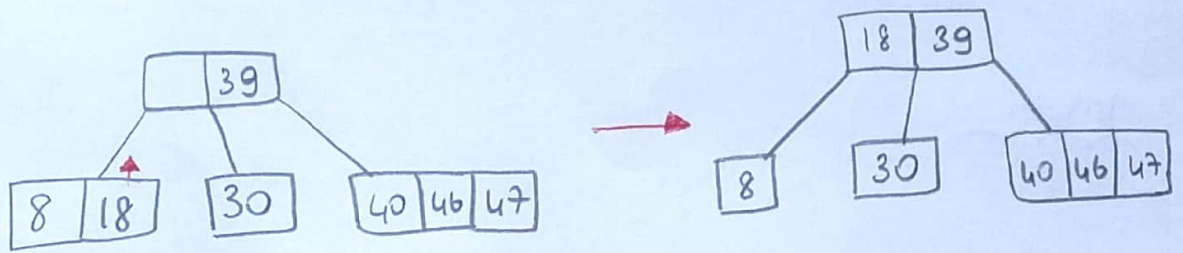


$$m = 4$$

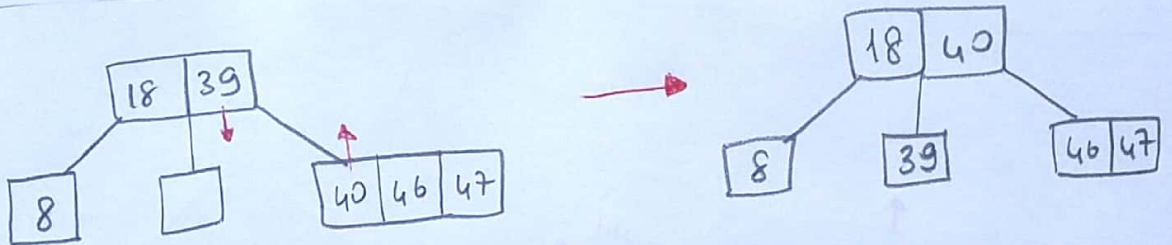
$$\text{max keys} = m - 1 = (3)$$

# Deletion

Remove 20



Remove 30



Remove 8

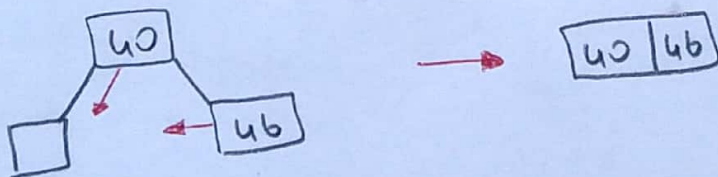


Remove 47

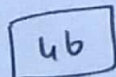


Remove 39

Remove 18



Remove 40



Remove 46