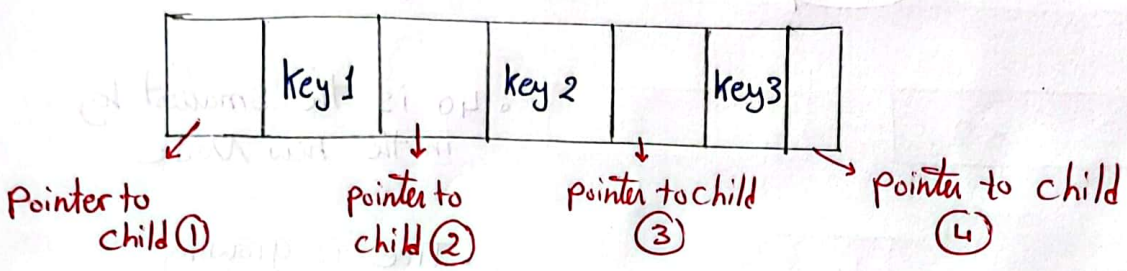
Insertion of B-tree Using time trade space offs

Structure of it

m-way → m is the number of pointer to child

example 4-way

keys = m-1 → 4-1 = 3 → key in each node



Rules

① We have to at least fill the half in every node
ex: if degree is (m) we have to fill $[m/2]$

② leaf nodes must be at same level

③ Root can have minimum ② children

④ We fill in Bottom up way
when we split we take the smallest key in the new node and then it becomes the parent

- It is similar to the search except it puts the new keys and splits into two nodes if needed

Insertion

Insert the following keys : 10, 20, 40, 50, 60, 70, 80, 30, 35, 5, 15

child pointers = 4 = m

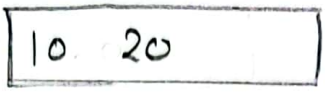
keys in node = 4-1 = 3

next page →

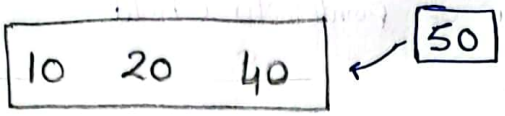
Nowran

(1) 

// ADD 10

(2) 

// ADD 20

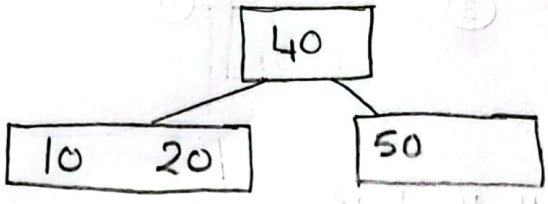
(3) 

no space for 50 so we will split this node into (2) → Create new one

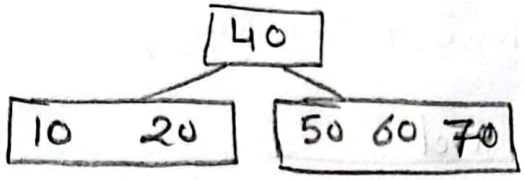
(4) 

• We will now Balance

• 40 is the smallest key in the new Node

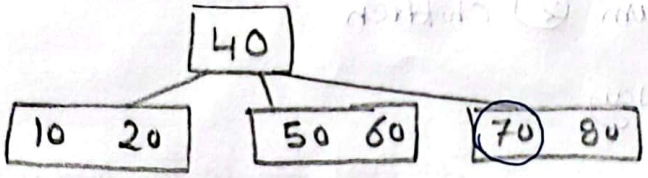


• Tree is growing upwards
• Splitting while creating a B tree

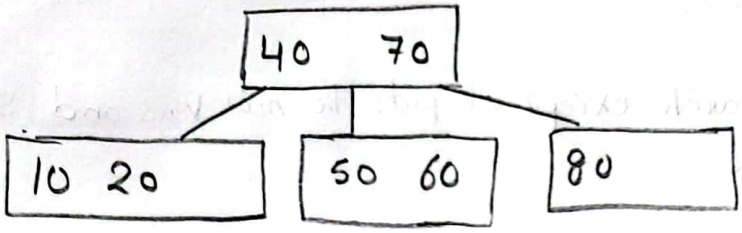
(5) 

// ADD 60, 70

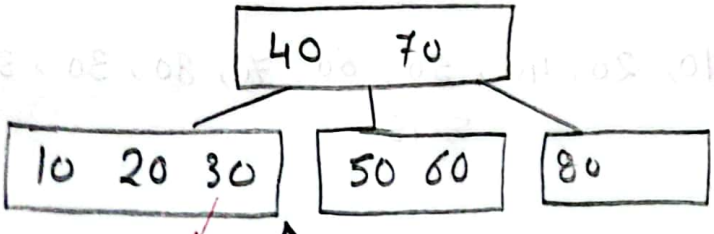
(80) no space



• 70 is the Smallest key in the New Node



• Balancing

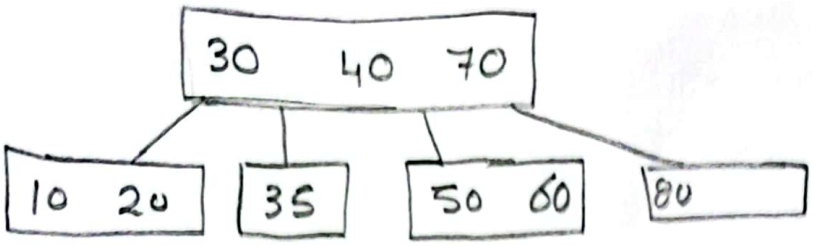
(6) 

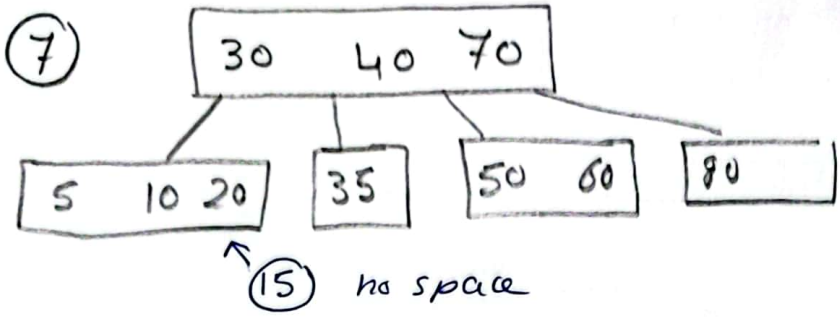
• // ADD 30

will be Smallest key in new node

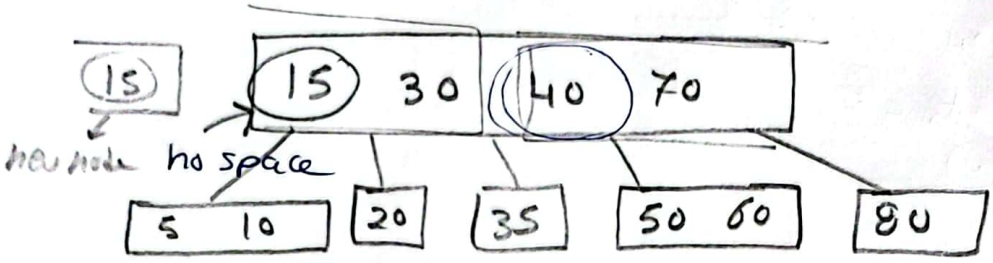
(35) no space

Nouran Hassan



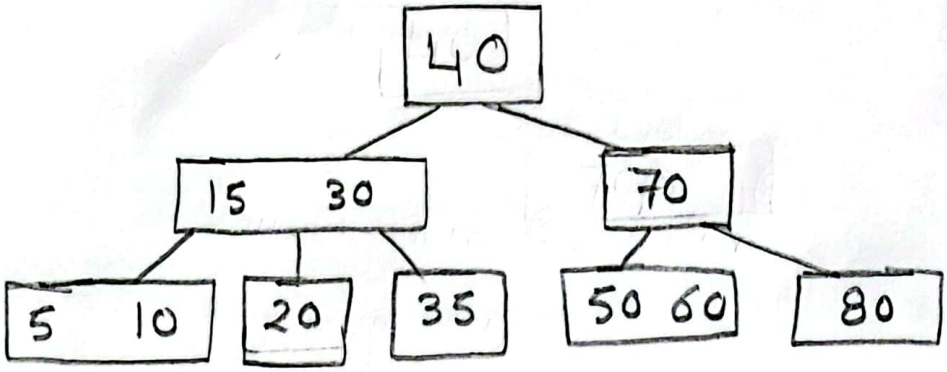


• ADD (5)



• Balancing

Final Result :



• All leafs are on the same level