

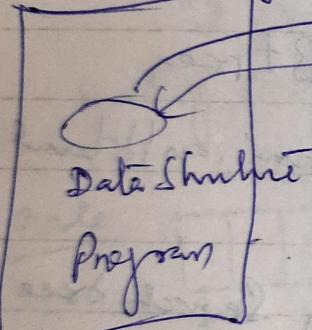
B Tree & B+ Tree:

①

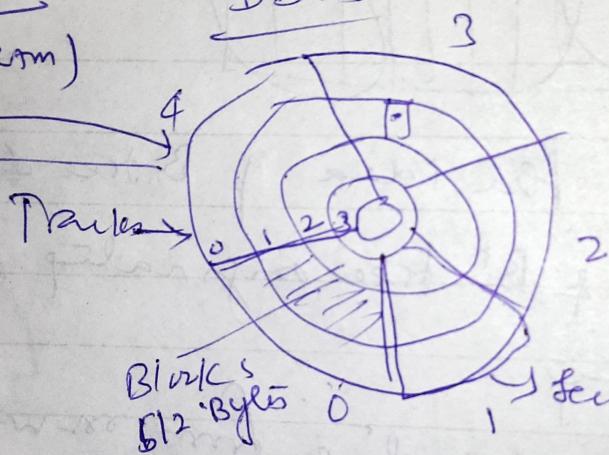
- disk structure
- how data is stored in disk
- indexing
- Multi-level indexing
- M way search trees
- B Trees, B+ Trees

Disk Structure:

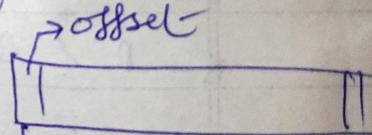
Main Memory (RAM)



DBMS



How many records
can store in a
block



Block add (Track no, sector no)

How data organized in files:

Employee

e_id → 10

name → 50

dept → 10

sechr → 8

add → 50

(Size) 128

No. of records / Block

$$= \frac{512}{128} = 4 \text{ records / Block}$$

$$\text{for 100 records} = \frac{100}{4} = 25 \text{ Block}$$

Select & store amp.

Index are used to
reduce the search time
of a record

e_id	name	dept	512
1	Ram	100	128 bytes
2	Lassi	100	→ 1 Block
3	:	100	
4		100	
5		100	
6		100	
7		100	
8		100	
9		100	
10		100	

100 records

$$\text{entry} = 10 \\ \text{pointer} = \frac{6}{16 \text{ bytes}}$$

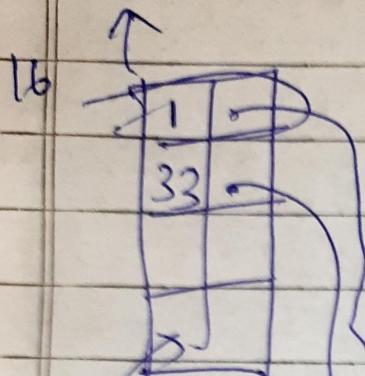
(2)

TNPL

பக்கம்:

நாள் : 11

Sparse index



32
blocks

index

Index	Pointer
1	1
2	2
3	3
4	4

32 entries

(i) $\rightarrow 1000$

Index	Value
1	Ram
2	Anil
3	Pri
4	Lam
5	

(2)

100 $\rightarrow 1000$

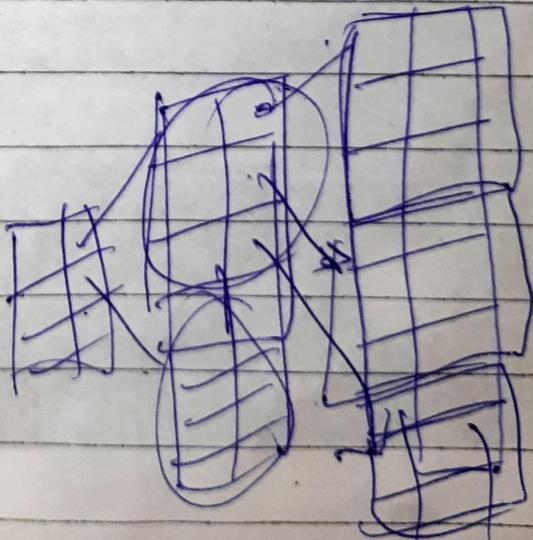
$$\text{No of entries for block} = \frac{512}{32} = 16$$

$$100 \text{ entries} = \frac{100}{16} = 6.25 \approx 7.$$

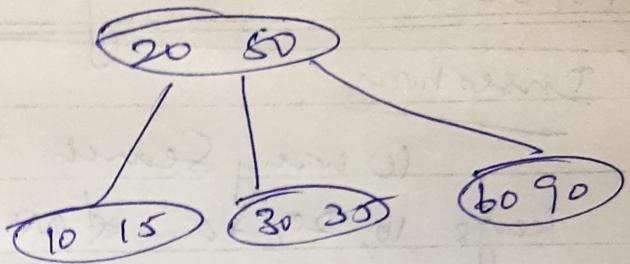
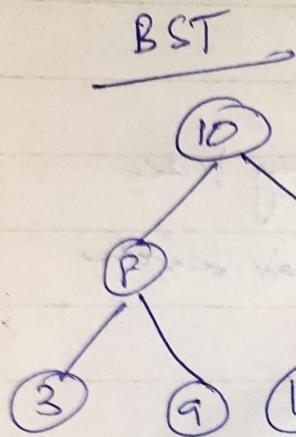
So 4+1 blocks instead 25 blocks

Multilevel indexing

originally a B+tree;



M-way Search tree: → Search tree $R_1 < R_2 < R_3$ (3)



2 keys

3 children

↓
3 way search tree

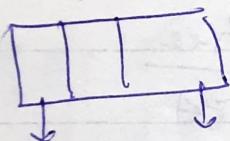
m-way ST

m -children \rightarrow degree of node

$m-1$ keys

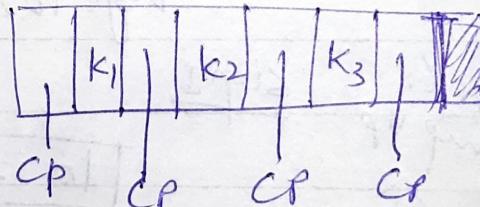
That's called binary tree

node



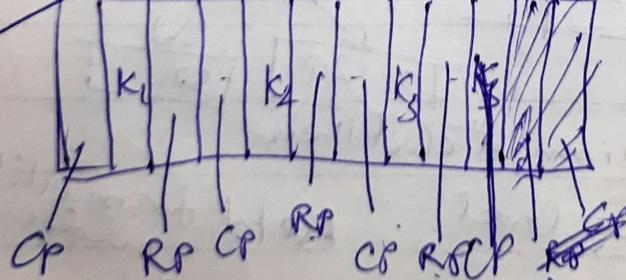
m-way ST

4 way ST



→ use this for preparing index table

Node Structure child ptr
Same for B-tree



where for multi-level indices

Index

rid	RF
1	RF
2	RF
3	RF

rid	RF
1	RF
2	RF
3	RF
4	RF
5	RF
6	RF
7	RF
8	RF
9	RF
10	RF

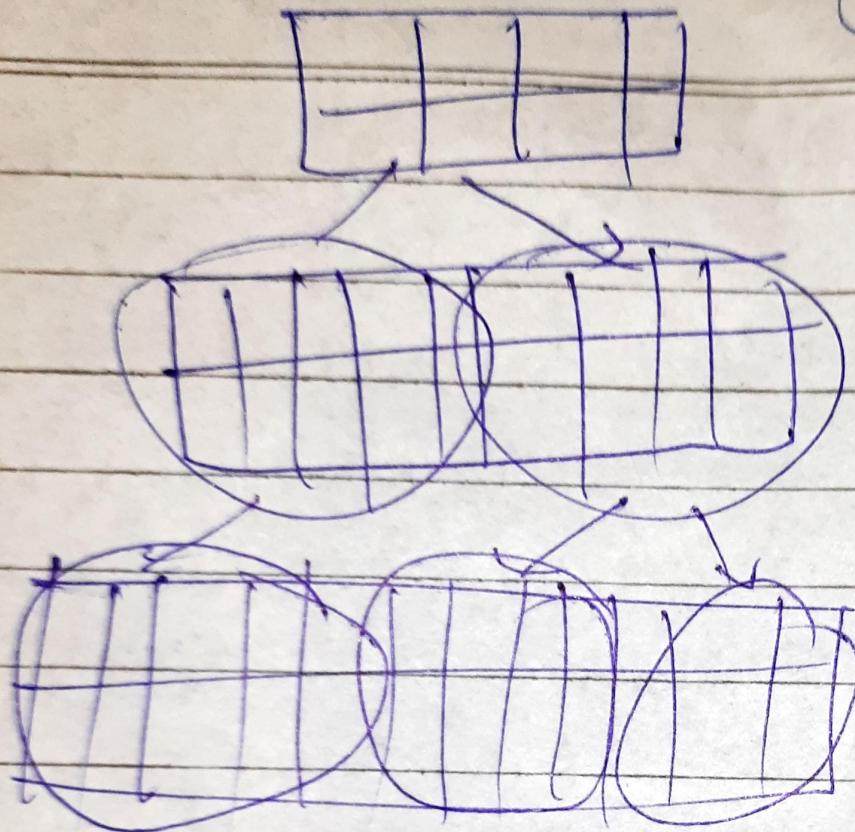
low

4

TNPL

பக்கம்:

நாள் : 11



Self managed
Multi level index

- * That is the idea of B tree & B⁺ tree
- * B tree & B⁺ tree originating from multi-level indexing.
- * D & B⁺ originating from m-way Search tree.
- *

5

Problem with m-way STree

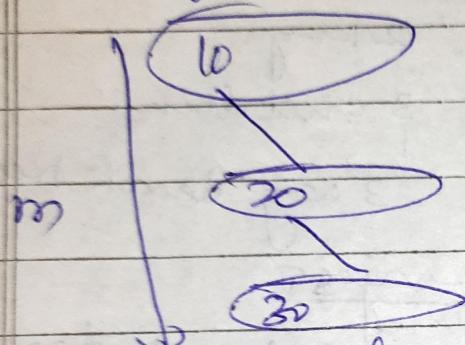
TNPL

பக்கம்:

நாள் : 11

Insertion

10 way Search tree
 keys 10, 20, 30 * no rules for M-way tree
 & similar to linear search



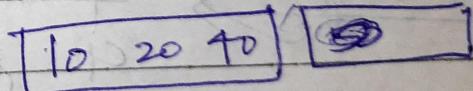
Midline for m-way tree \rightarrow B tree

B Tree - is an m-way tree with the same * rules.
~~ways to 20 30 ... n~~

B tree
 $m=4$

keys 10, 20, 30, 40

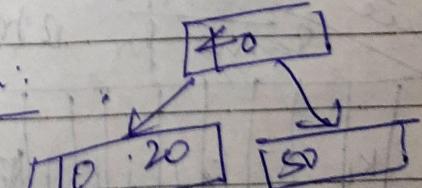
tree is
group up Step 1



10, 20, 30, 40

splitting done

Step 2:



Rule

1. $\lceil \frac{m}{2} \rceil$ children

2. Root can have minimum 2 children

3. All leaf at same level

4. Bottom up

useful for multi-level
index -

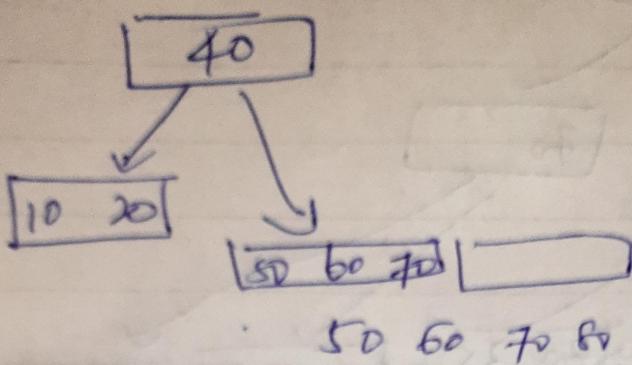
steps

keys inserted

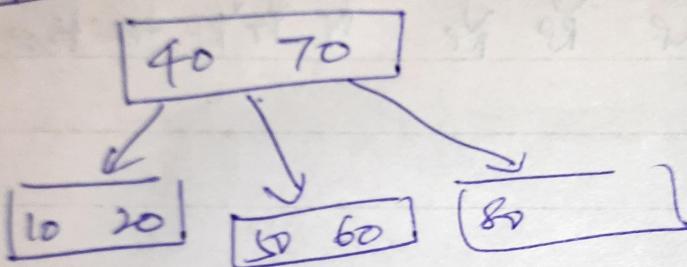
10, 20, 40, 50, 60, 70, 80, 30,

6

35, 5, 15



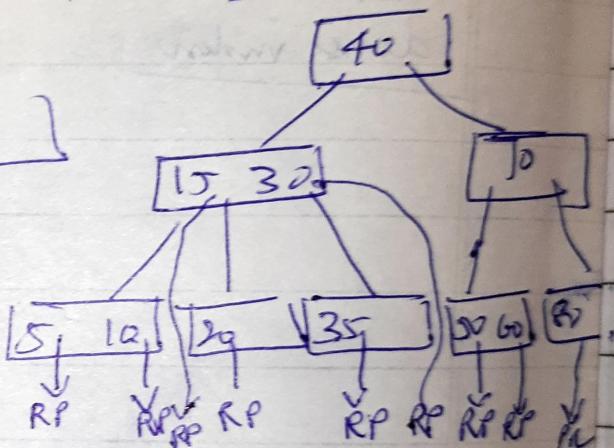
step 4:



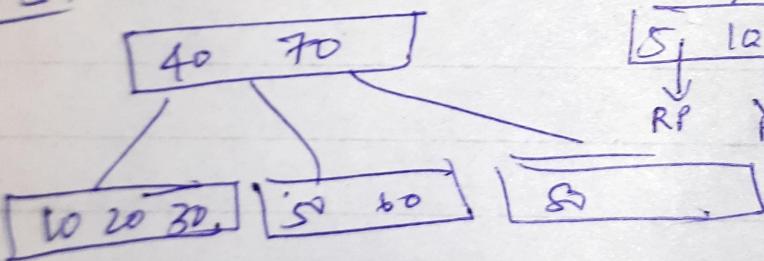
Same as m level merging

Step 9

3 level



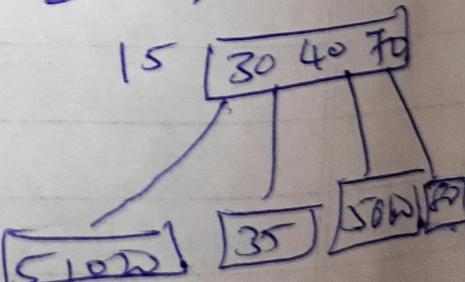
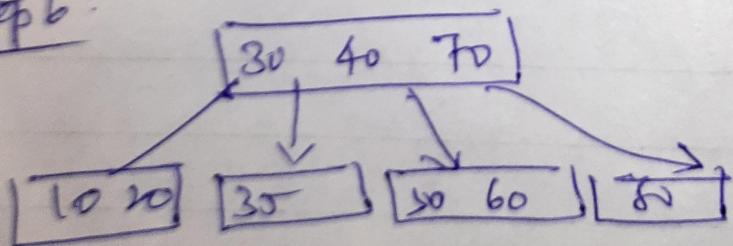
Step 5:



10, 20, 30, 35

→ no space

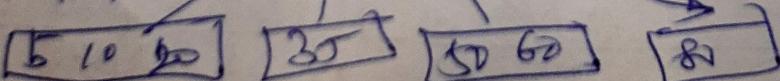
Step 6:

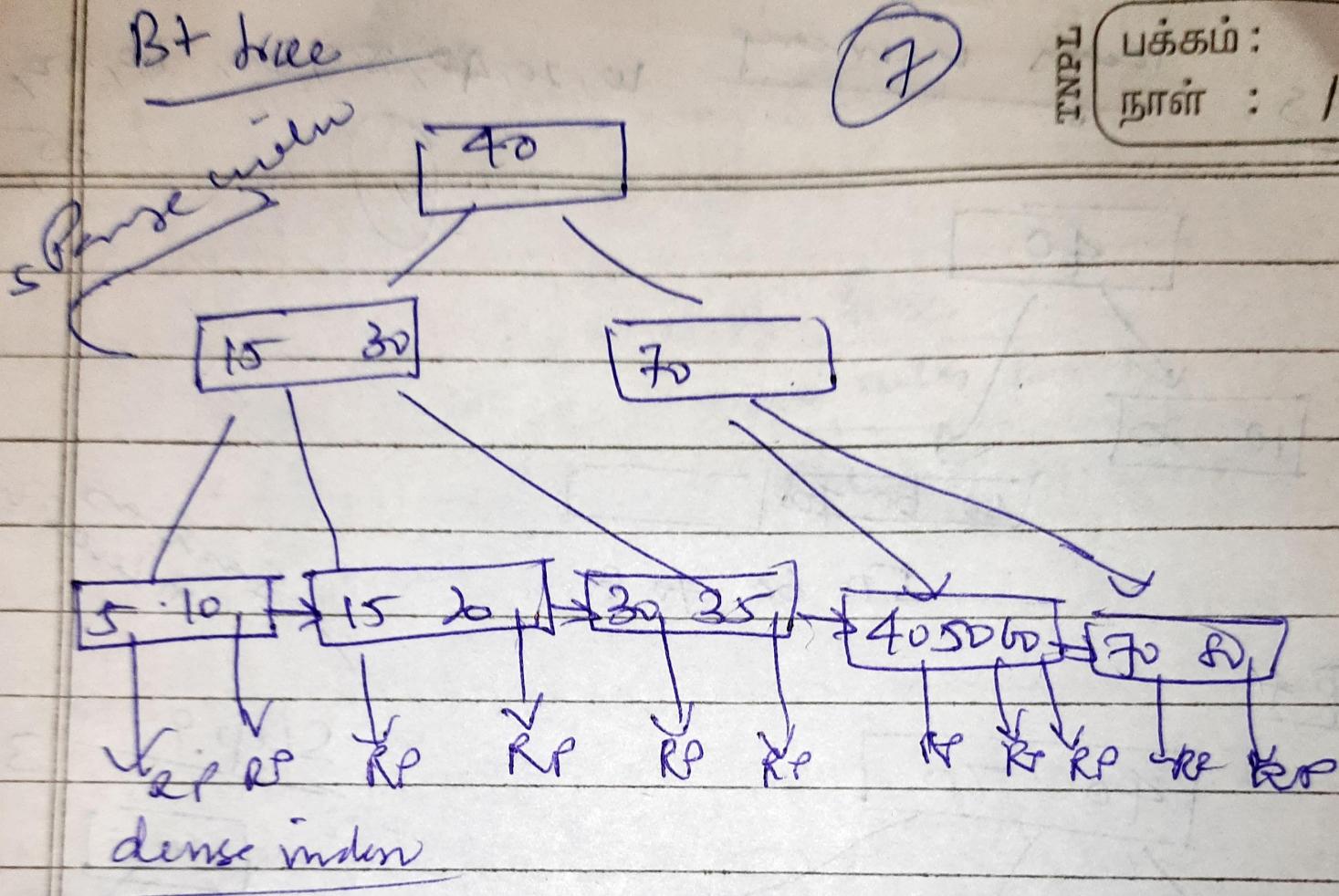


Step 7:

5, 10, 20, 30

30 40 70





(7)

TNPL

பக்கம்:
நாள் : 1 / 1