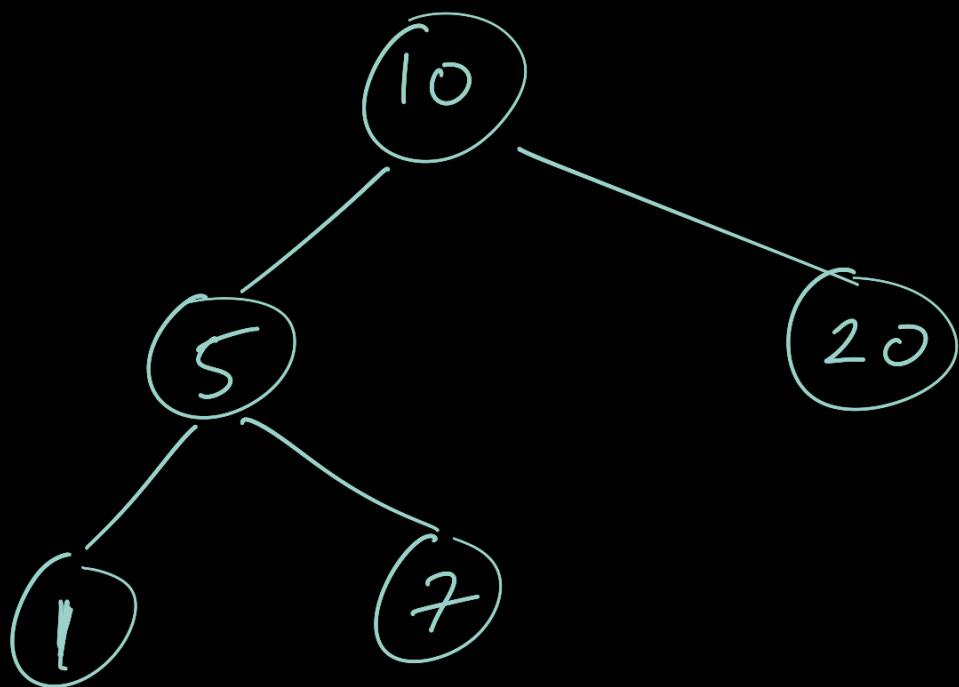


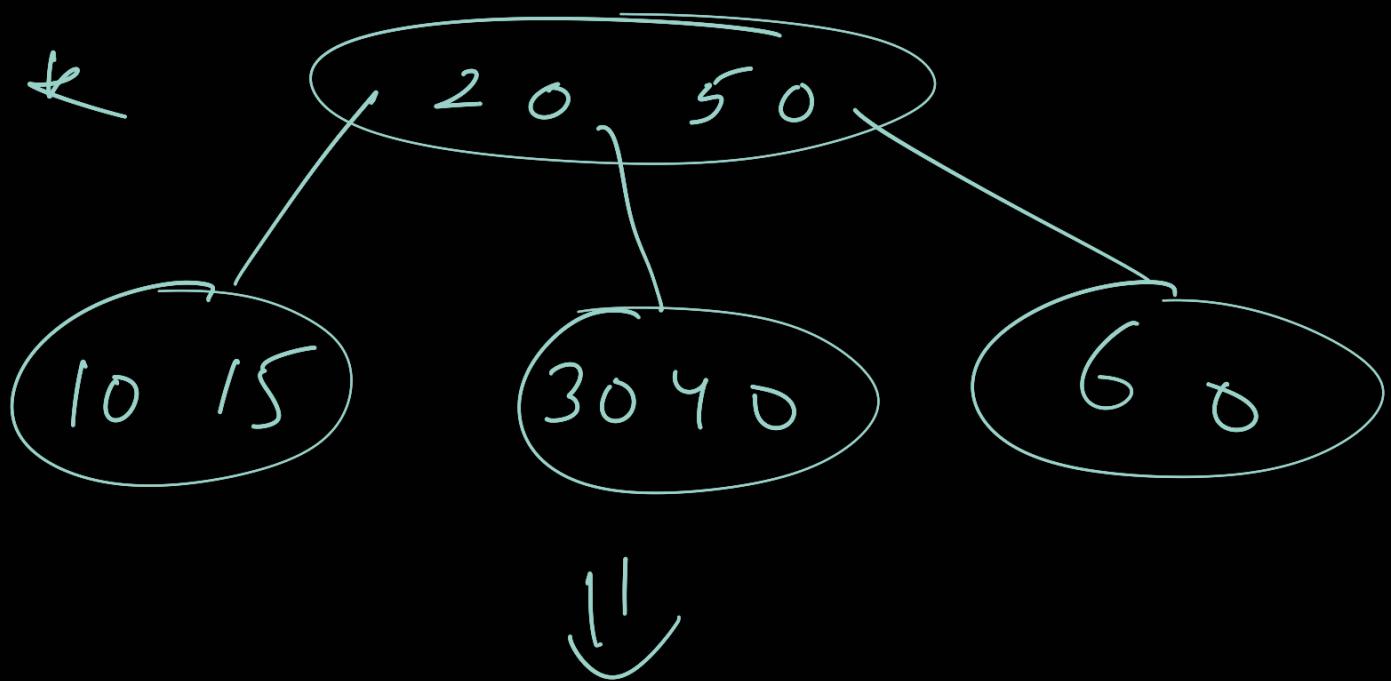
BST



* Each node can have 1 key only

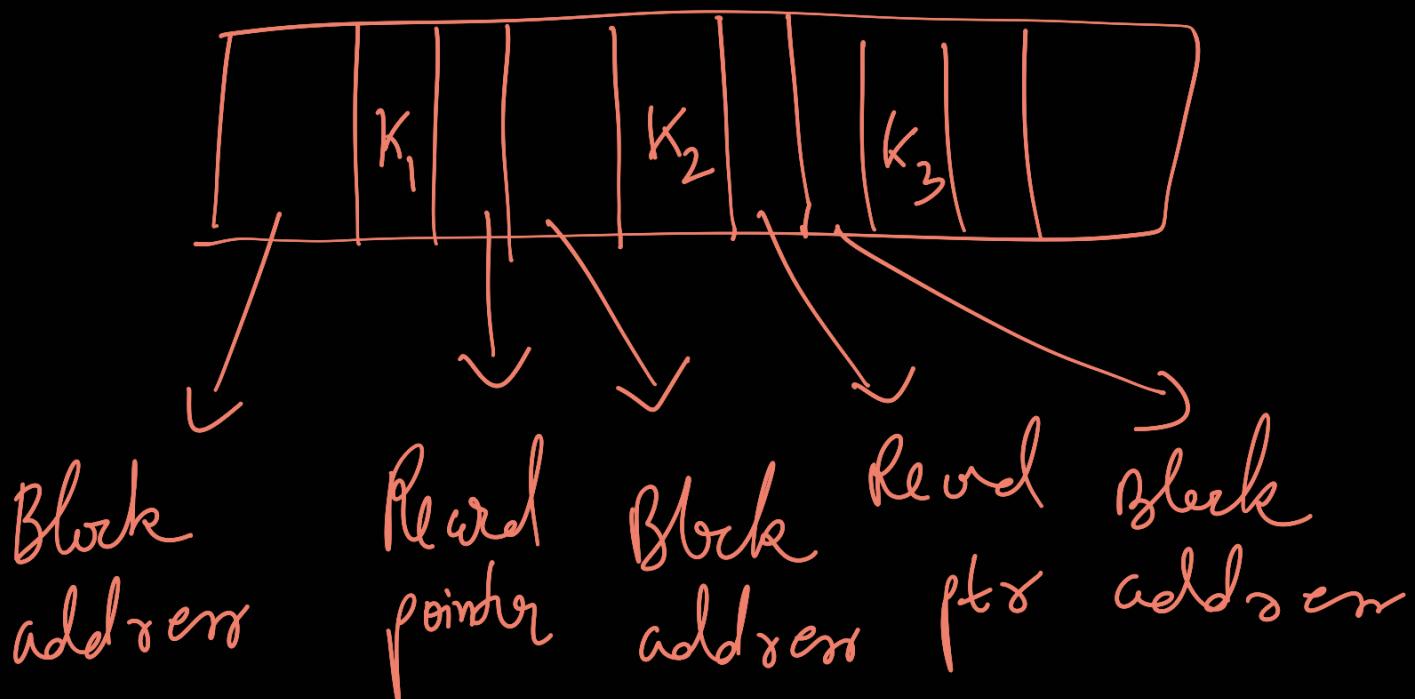


M-way search tree

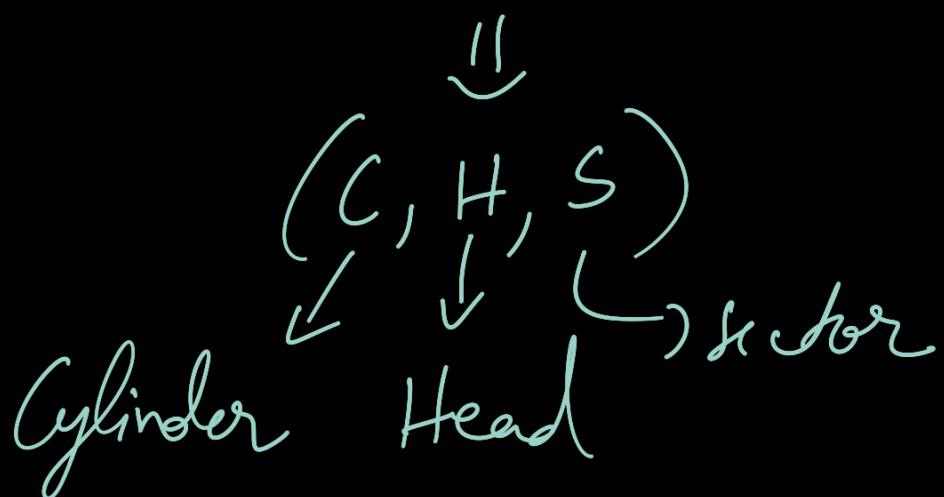


* B, B⁺ tree are M-way search tree with some conditions.

B-tree index file



* Rewind pointer = Block + Byt offset address



- * Each Node of tree generally equal to size of HOD block
- * So, order of tree is determined by HOD block size
 - ↳ Generally $\rightarrow 50 \rightarrow 200$

Order = Max. children children
a node can have

Order \uparrow = Levels \downarrow = Less
HOD block
access
= Less I/O time

① Root node

- ↳ Min children = 2 (Binary)
- ↳ More " = order = p of tree

② Internal node

- ↳ Min $\lceil \frac{p}{2} \rceil$ children
- ↳ More p children

③ Leaf nodes → all at same level

→ No Block ptr only record pointer

- * Let consider a empty table with index created on some column
 - * Table empty = index empty
 - * Now some record got inserted
 - ↓
 - Index has to updated
 - ↑
 - Tree updation/creation starts
- E.g -> Insert 10 (consider order = 4)

10

Insert 20

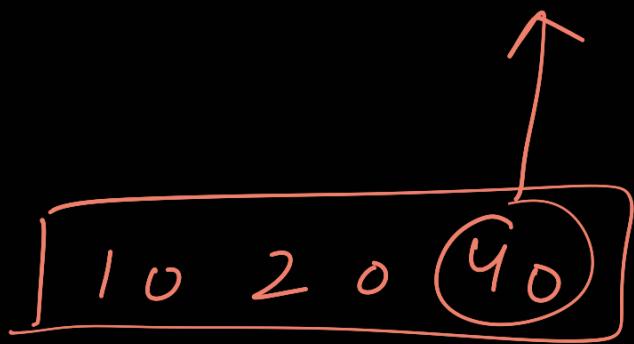


Insert 40



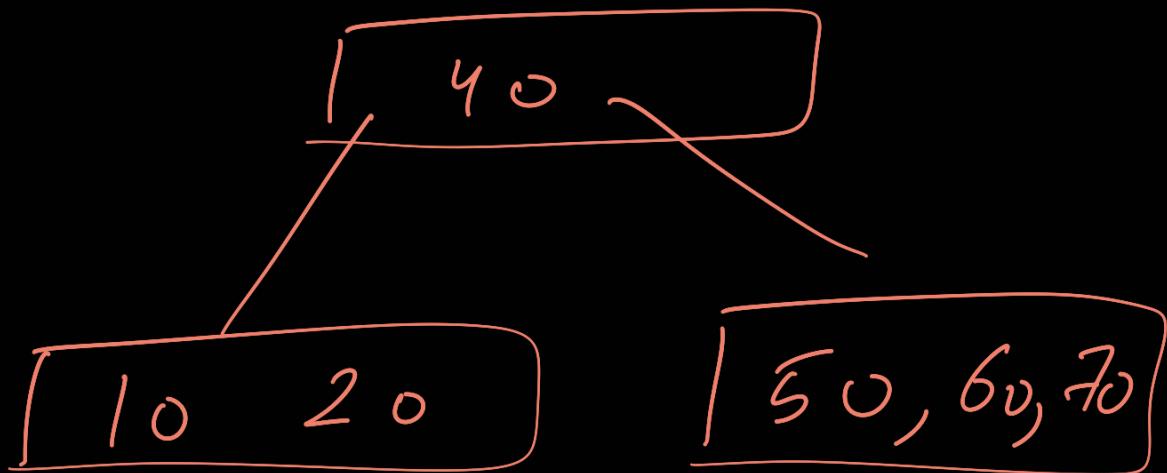
Insert 50 (order = 4, log = 3(n))

Split node

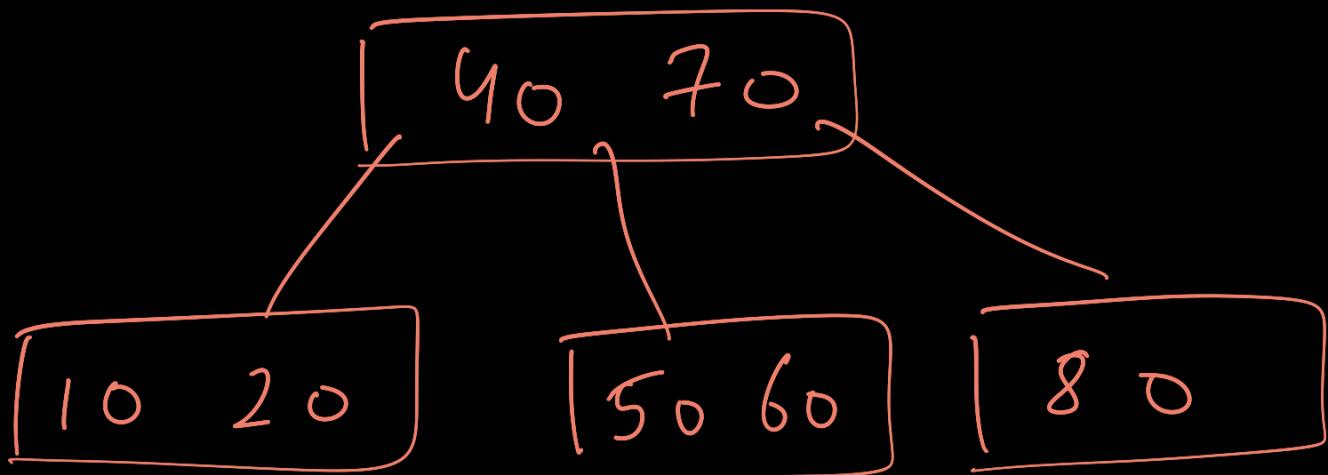
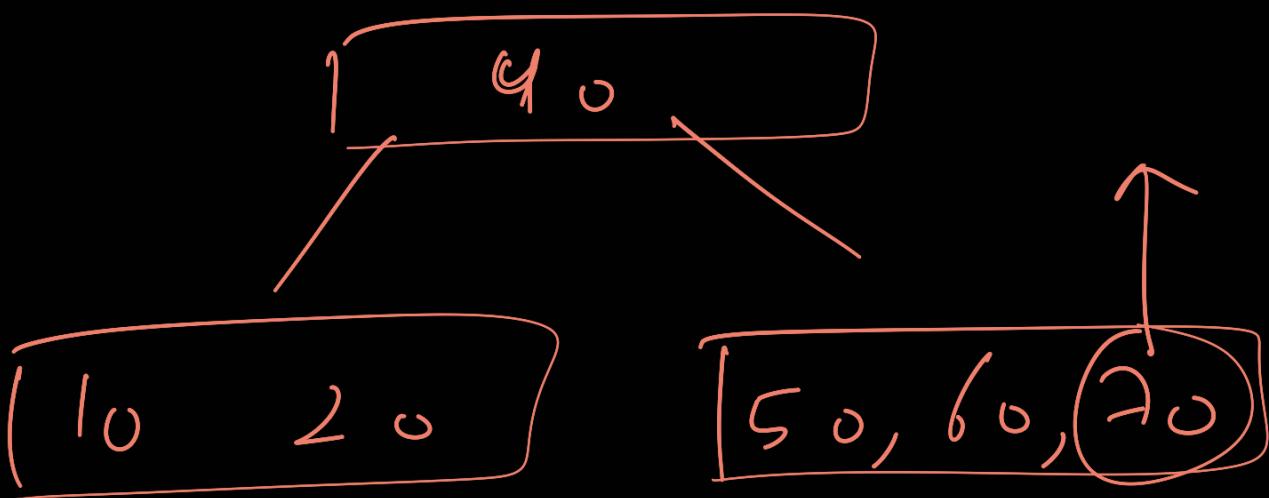


These are
HDD blocks

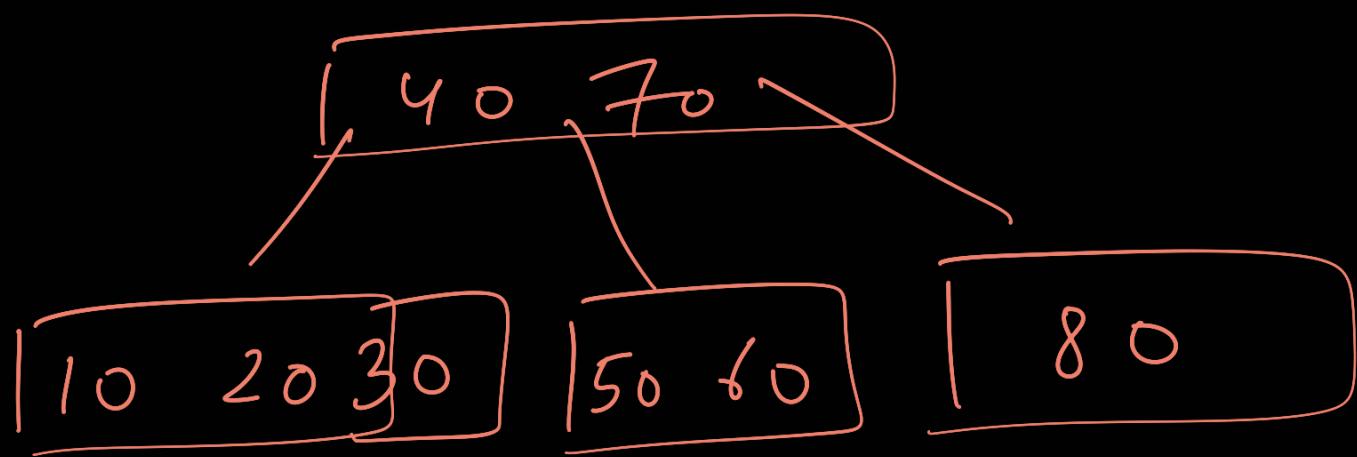
Insert 60, 70



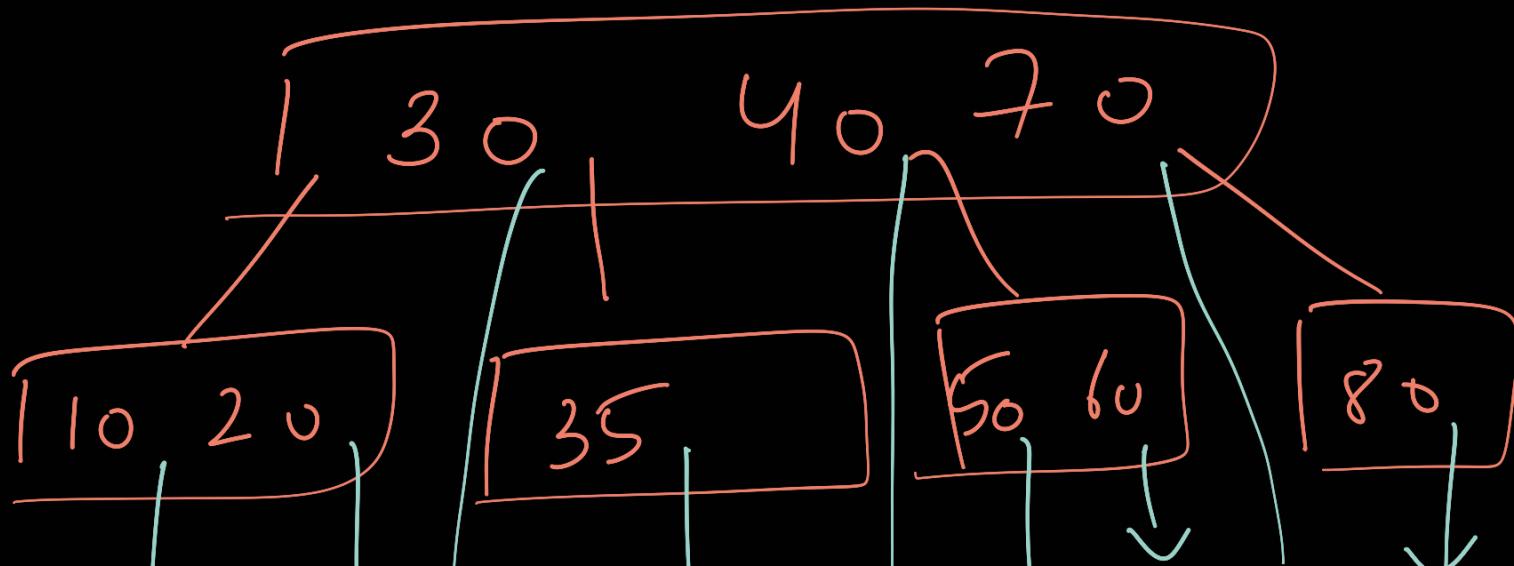
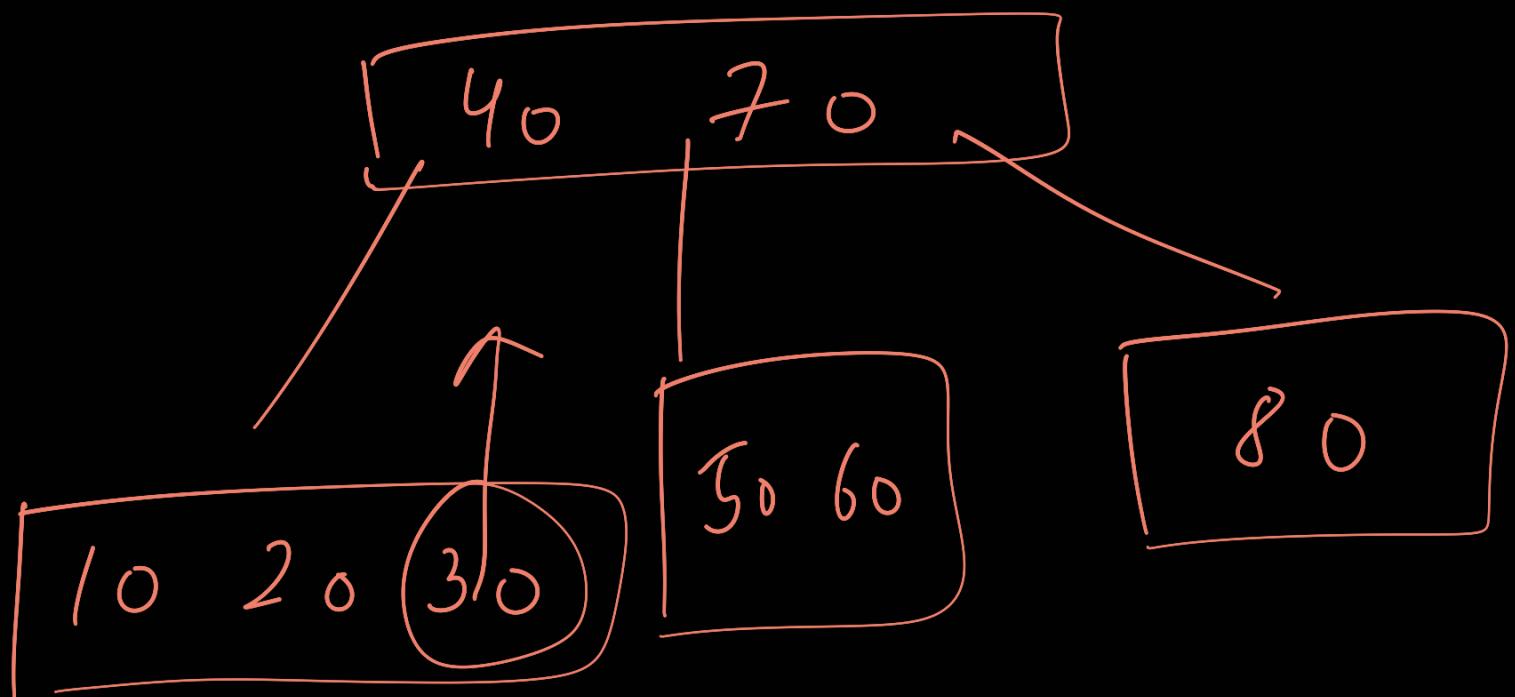
Insert 80

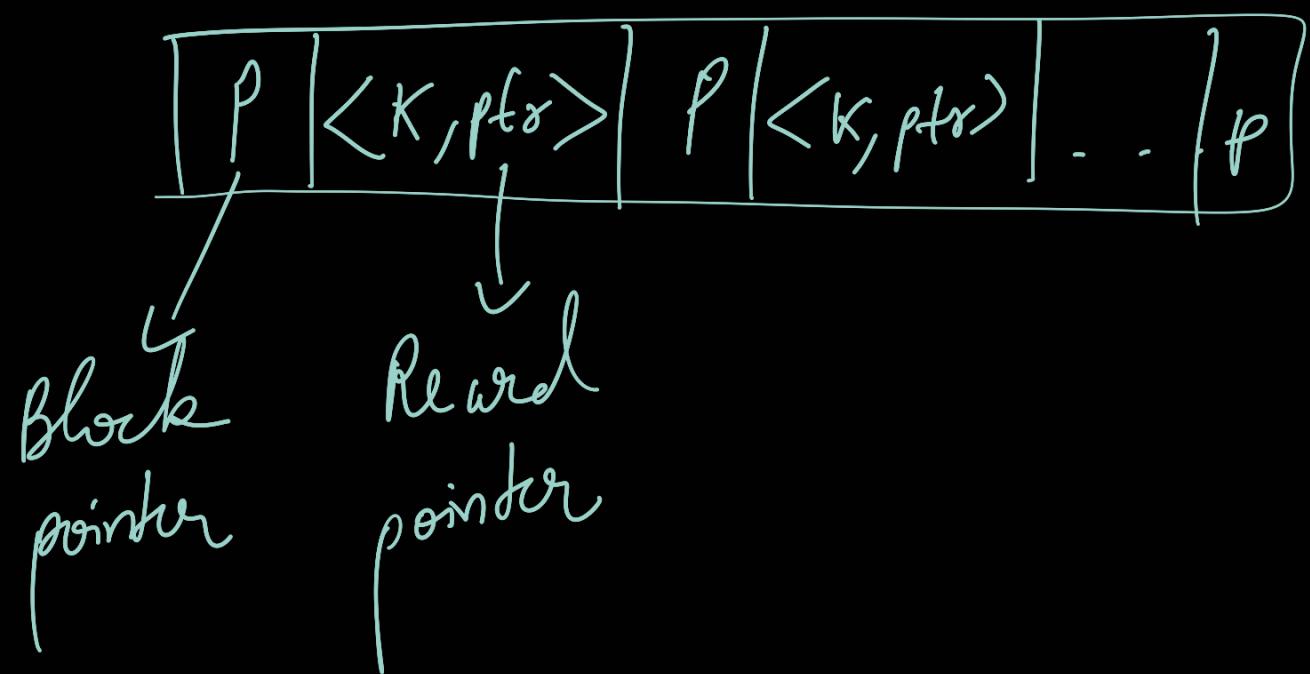
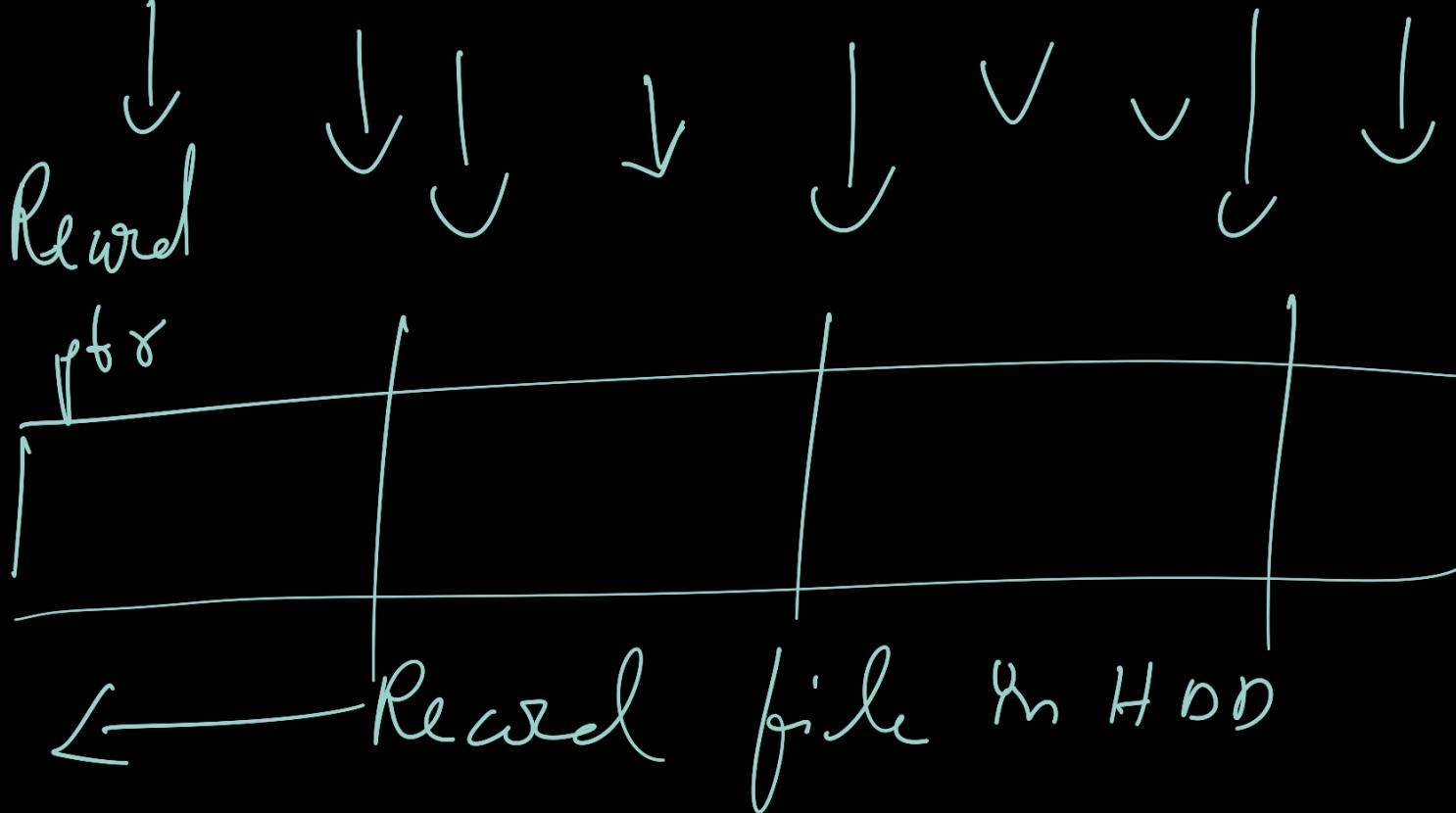


Insert 30



Insert 35





* If a node is not full,
some space of HDD sector/block
is wasted

B⁺ - Index file

- * Useful for Range Queries
"BETWEEN"
 - * Leaf and non-leaf have different structure.
- ① Non-leaf nodes don't have Record pointer
 - ② Leaf node has all index entries, non-leaf may contain duplicates

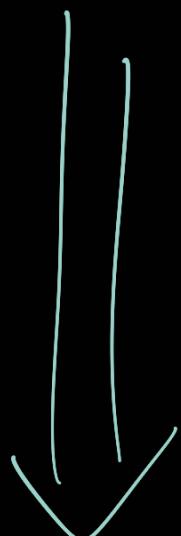
P	K_1	$P.$	K_2
-----	-------	------	-------

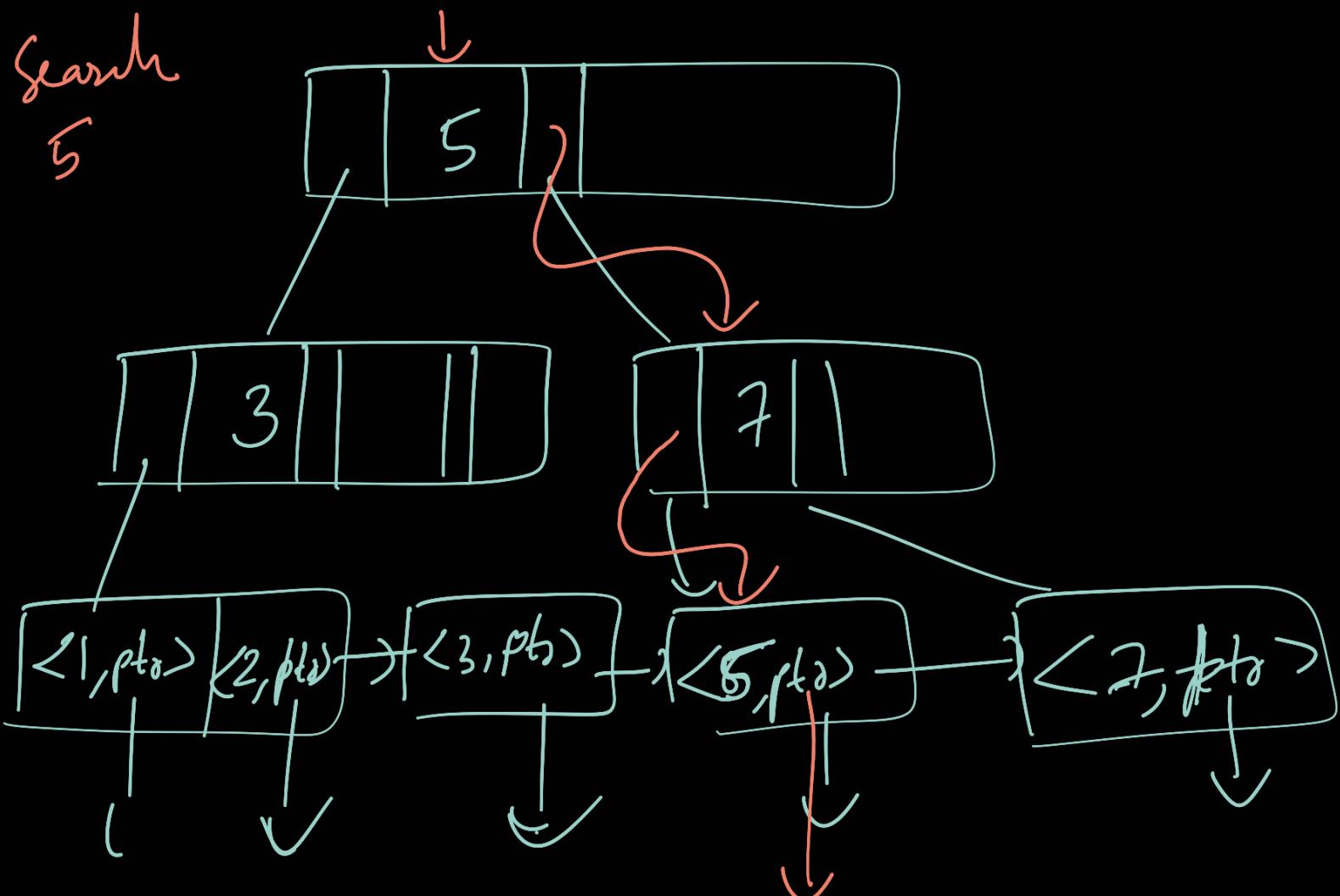
\rightarrow Non-leaf

Block
pointer Key

$\langle K_1, R_1 \rangle$	$\langle K_2, R_2 \rangle$	\dots
----------------------------	----------------------------	---------

Key Reward
 pointer

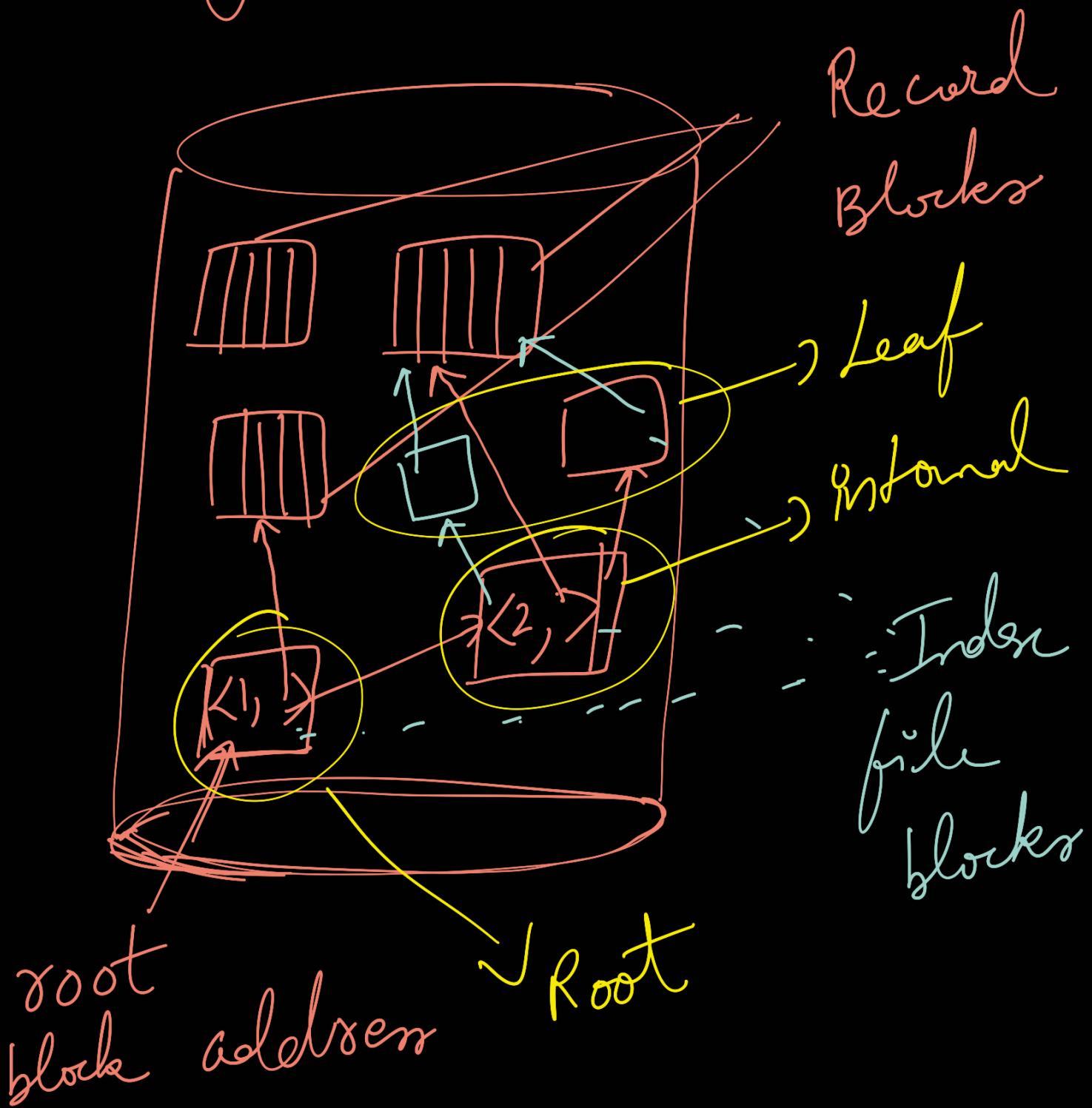




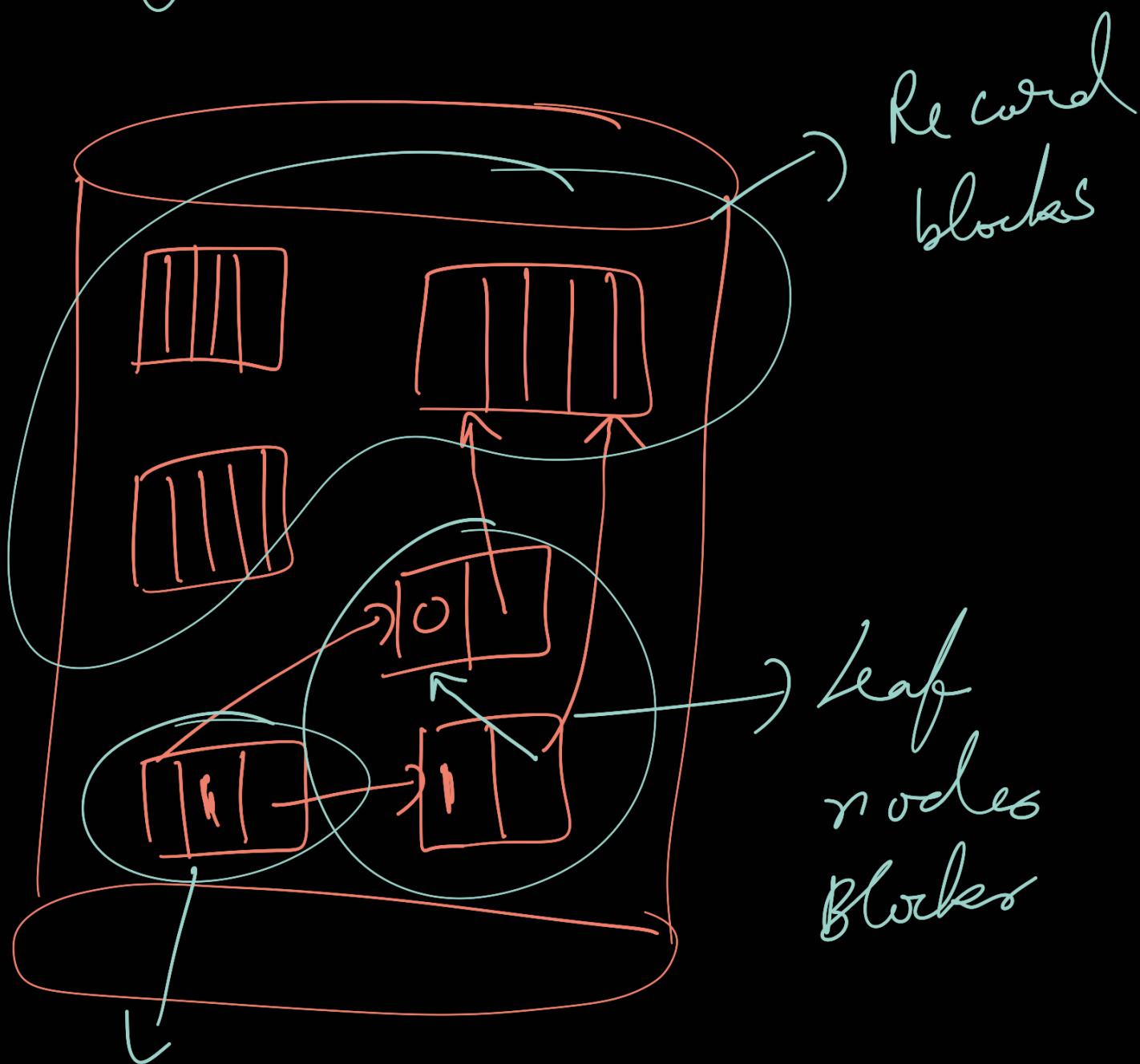
- * On first HDD Block access for root \rightarrow we got 5, but second pointer is not present, so we still have to go to leaf.
- * Leaf nodes contain pointer to next node, Range queries can be

optimised.

Visualising B-tree index file



Visualising B^+ tree



root
block

Record
blocks

leaf
nodes
blocks