COMPUTER SCIENCE

Database Management System

File Org. & Indexing

Lecture_3

Vijay Agarwal sir

TOPICS TO BE COVERED

01 File Structure & Indexing

02 B Tree & B+ Tree

Indexing

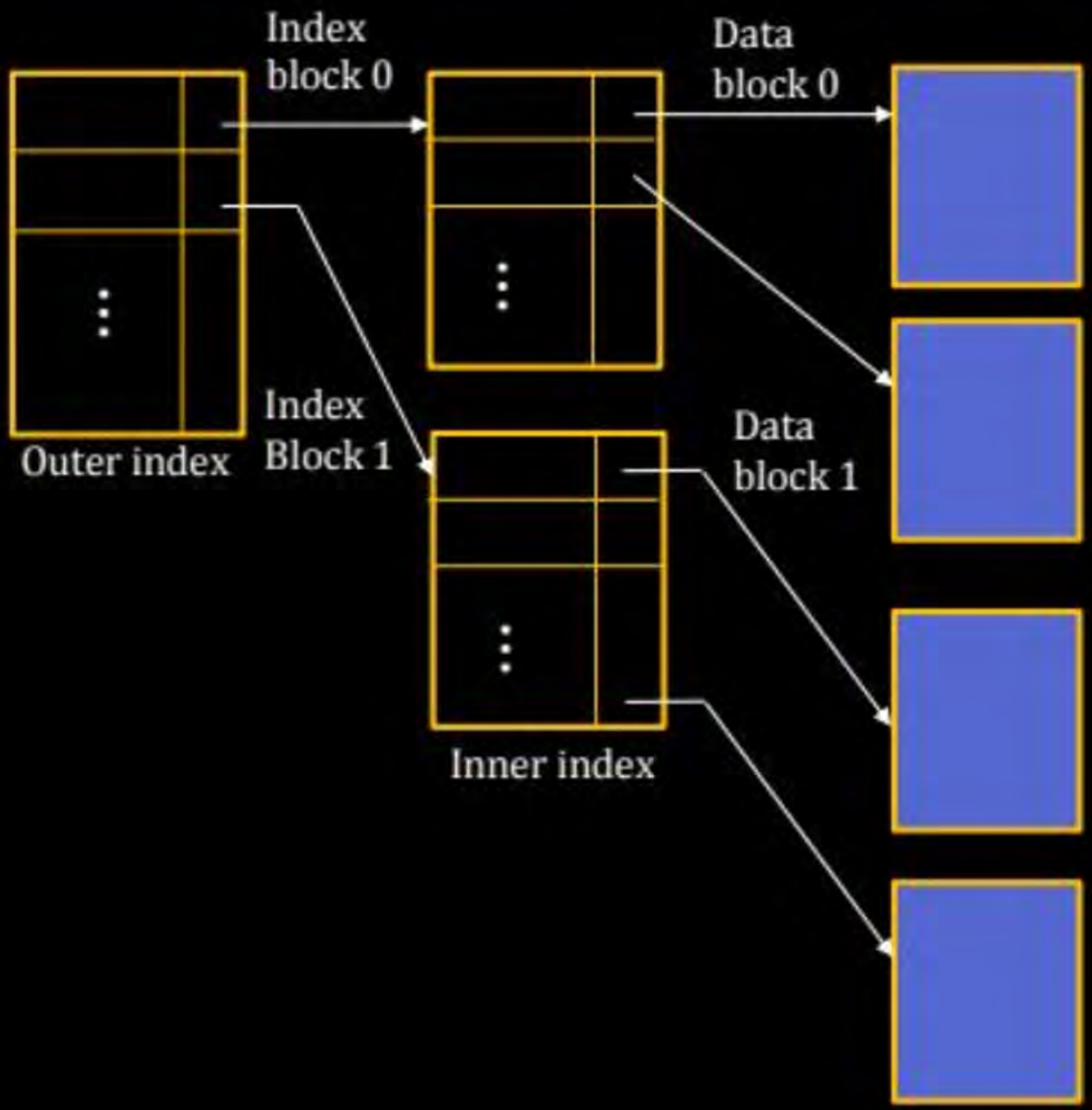
1. P.I
2. C.I
3. S.I

Multilevel Indexing

Multilevel Index

- If index does not fit in memory, access becomes expensive.
- Solution: treat index kept on disk as a sequential file and construct a sparse index on it.
  - outer index - a sparse index of the basic index
  - inner index - the basic index file
- If even outer index is too large to fit in main memory, yet another level of index can be created, and so on.
- Indices at all levels must be updated on insertion or deletion from the file.

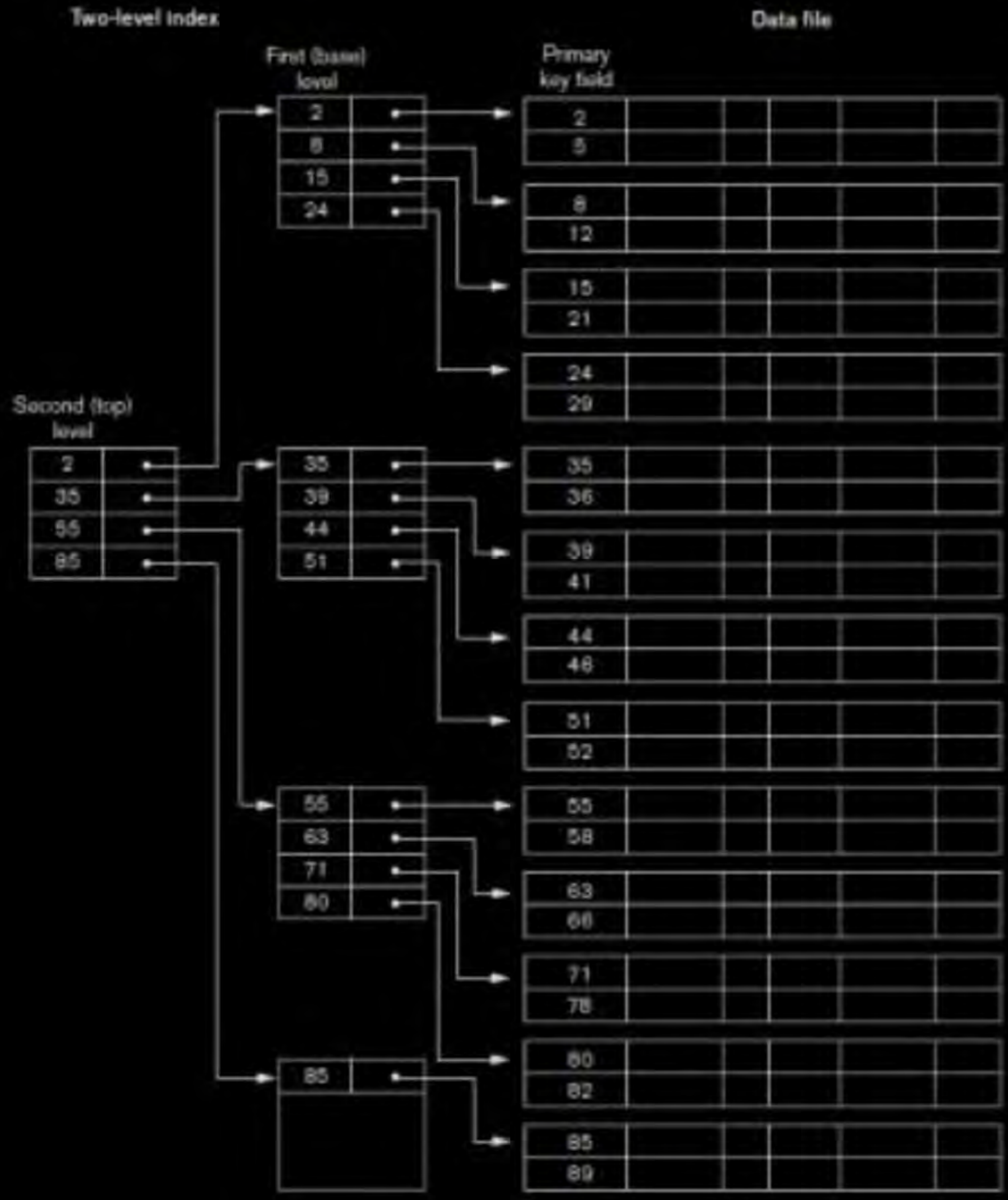
Multilevel Index (Contd..)



Multilevel Indexes

- Designed to greatly reduce remaining search space as search is conducted
- Index file
  - Considered first (or base level) of a multilevel index
- Second level
  - Primary index to the first level
- Third level
  - Primary index to the second level

A two-level primary index resembling ISAM (indexed sequential access method) organization



NOTE: We can Repeat the above process until index entries fit into One Block.

NOTE: If there are n level in multilevel index then the number of Block Access to search for a record = $n + 1$

(at each level One Index Block + 1 Data Block)

Q.3 Find the average number of block access required to search for a record if multilevel Index is created on the Data file of Question 2.

Block factor of Index file = 68 Index entries per Block

I^st^ Level: Total number of Index Block = 442 Index Block

II^nd^ Level: number of Index Records (entries) = 442 (SPARSE number of I^st^ level block) & Block factor = 68 Index entries for block

Total number Index Block = $\left\lceil \frac{442}{68} \right\rceil = 7$ Index Block

III^rd^ Level: Number of Index Record = 7 (number of 2^nd^ level block)

Total number of index Block = $\left\lceil \frac{7}{68} \right\rceil = 1$

Average Number of block Access = $1 + 1 + 1 + 1 = 4$

Q.1 A clustering index is defined on the fields which are of type

[GATE-2008 : 1 Mark]

(A) Non-key and ordering

(B) Non-key and non-ordering

(C) key and ordering

(D) key and non-ordering

Q.2 Consider a file of 16384 records. Each record is 32 bytes long and its key field is of size 6 bytes. The file is orderd on a non-key field, and the file organization is unspanned. The file is stored in a file system with block size 1024 bytes, and the size of block pointer is 10bytes. If the secondary index is built on the key field of the file, and a multilevel index scheme is used to store the secondary index, the number of first-level and second-level block in the multilevel index are respectively [GATE-2008 : 2 Marks]

(A) 8 and 0

(B) 128 and 6

(C) 256 and 4

(D) 512 and 5

Tree

Binary Tree [At Most Degree = 2] 0,1,2

BST.

AVL

!

B Tree Definition

Balanced Tree

DSA

ORDER : P

Max. element = $P-1$

Min. element = $\lceil P/2 \rceil - 1$

BP

ORDER : P

Max keys = $P-1$

Min. Key = $\lceil P/2 \rceil - 1$

eg ORDER : 5

Max. keys = $5-1 = 4$

Minimum keys = $\lceil 5/2 \rceil - 1 = 3-1 = 2$

Dynamic Multi Level Indexing.

B Tree

ORDER : P

B_p : P

Key = P−1

R_p = P−1

B_1	$K_1 R_1$	B_2	$K_2 R_2$	B_3	$K_3 R_3$	. . . .	B_{p-1}	$K_{p-1} R_{p-1}$	B_p

B Tree Definition

Balanced Tree

① ORDER : 2

$B_P = 2$

$key = 1$

$R_P = 1$

B_1	$K_1 R_1$	B_2

② ORDER : 3

$B_P = 3$

$keys = 2$

$R_P = 2$

B_1	$K_1 R_1$	B_2	$K_2 R_2$	B_3

③ ORDER : 4

$B_P = 4$

$key = 3$

$R_P = 3$

B_1	$K_1 R_1$	B_2	$K_2 R_2$	B_3	$K_3 R_3$	B_4

④ ORDER : 5

$B_P = 5$

$key = 4$

$R_P = 4$

B_1	$K_1 R_1$	B_2	$K_2 R_2$	B_3	$K_3 R_3$	B_4	$K_4 R_4$	B_5

⑤ ORDER : 6

$B_P = 6$

$key = 5$

$R_P = 5$

B_1	$K_1 R_1$	B_2	$K_2 R_2$	B_3	$K_3 R_3$	B_4	$K_4 R_4$	B_5	$K_5 R_5$	B_6

⑥ ORDER : P

$B_P = P$

$keys = P-1$

$R_P = P-1$

B Tree Definition

Balanced Tree

DSA

ORDER : P

$Bp = P$

Max. keys $= P-1$

$Rp = P-1$

Min keys $= \lceil P/2 \rceil - 1$

ORDER : 5

$Bp = 5$

max key $= 4$

min key $= \lceil 5/2 \rceil - 1 = 2$

eg 1, 5, 3, 7, 9, 11

1	3	5	7	← 9

B Tree Condition Violate

1	3	5	7	9

B Tree Definition

Balanced Tree

DBMS

ORDER : P

Bp = P

Max. keys = P−1

Rp = P−1

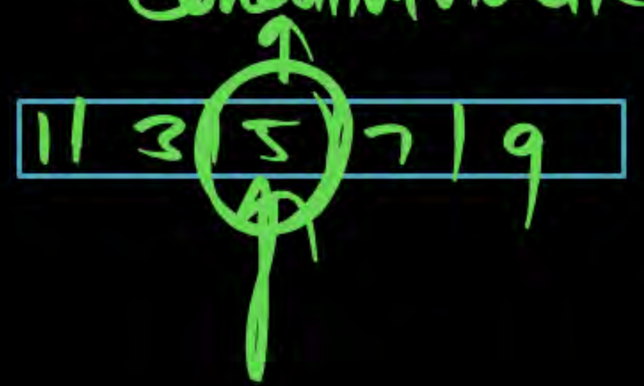
Min keys = $\lceil P/2 \rceil - 1$

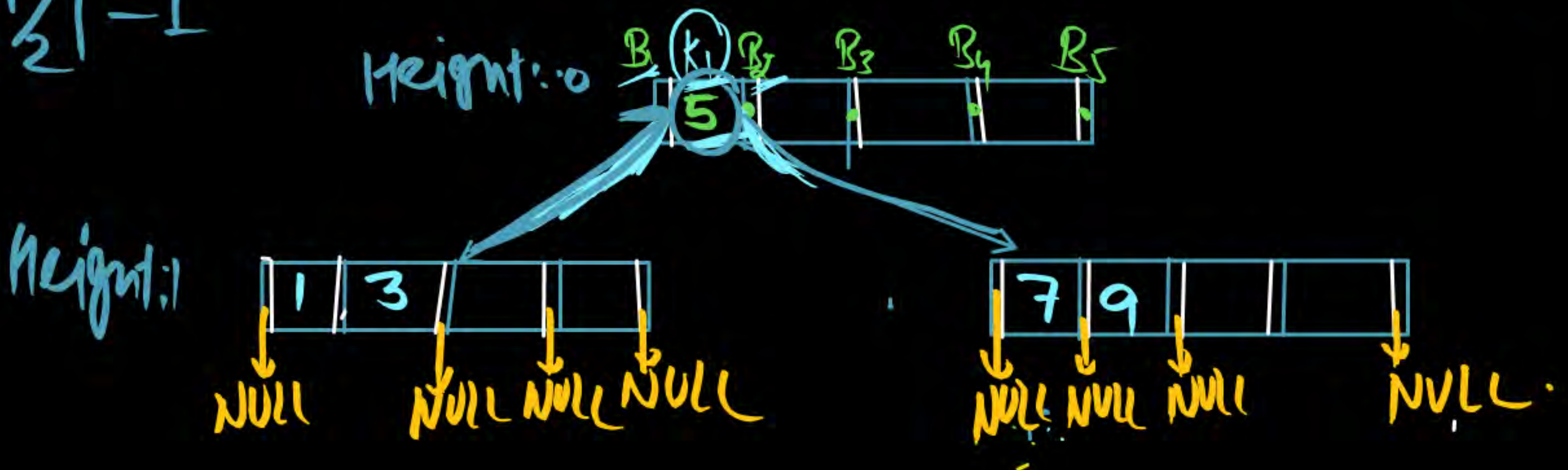
ORDER : 5
Bp = 5
key = 4
Rp = 4

ORDER : 5

Bp = 5
max key = 4
min key = $\lceil 5/2 \rceil - 1 = 2$

eg 1, 5, 3, 7, 9, 11





B Tree Definition

Balanced Tree

DSA ORDER : 4

Max key = $4-1=3$

Min key = $\lceil 4/2 \rceil - 1 \Rightarrow 2-1 = 1$

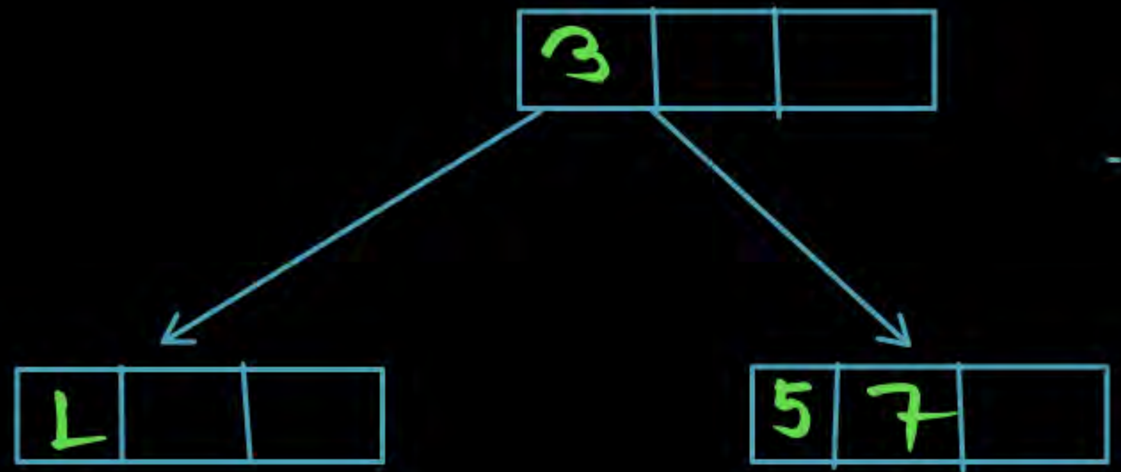
1, 3, 5, 7, 9, 11

1	3	5

← '7'

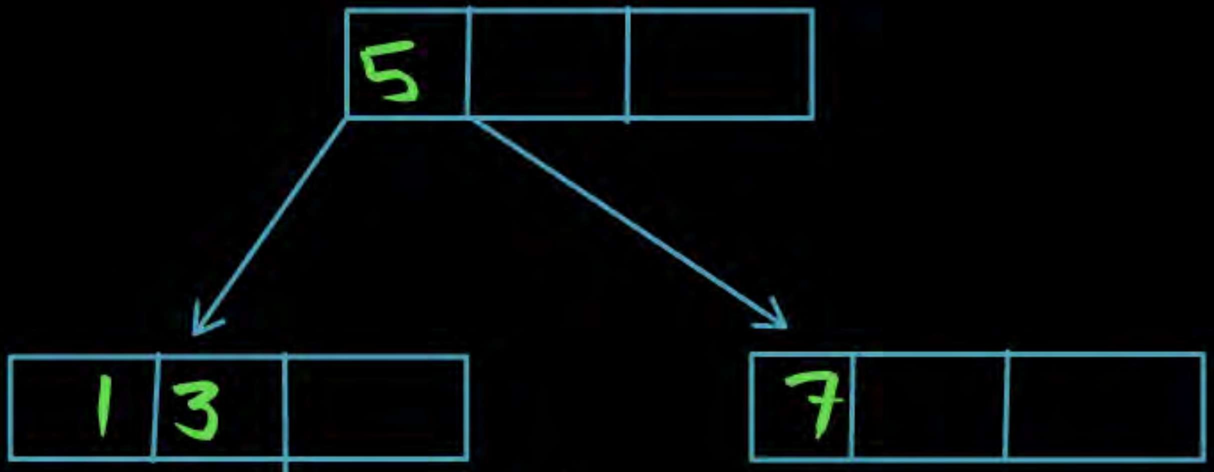
B Tree Defination violate

1	3,	5,	7



Right Biasing

OR



Left Biasing

B Tree Definition

Balanced Tree

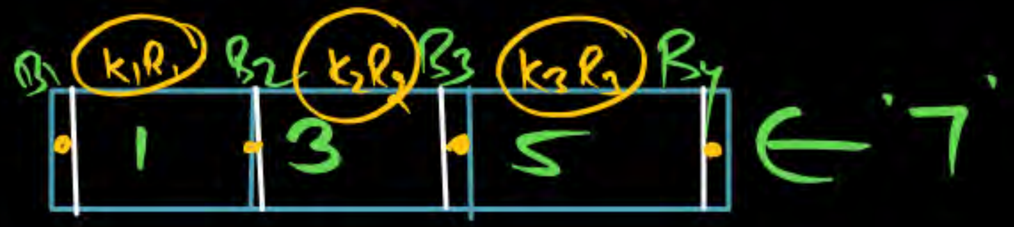
DBMS ORDER: 4

Max key $= 4-1 = 3$

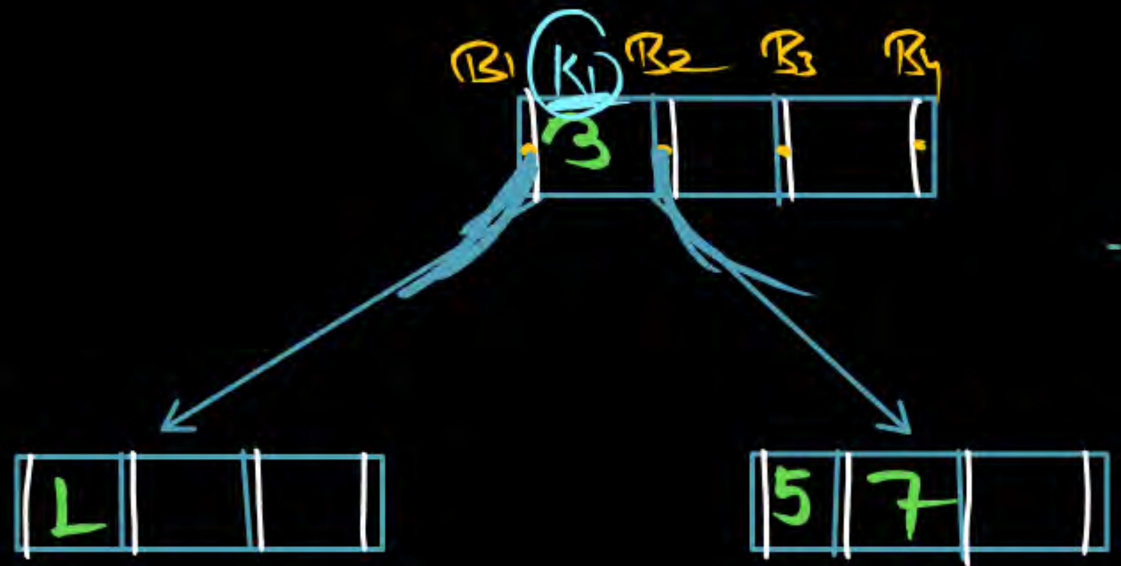
Min key $= \lceil 4/2 \rceil - 1 \Rightarrow 2-1 = 1$

ORDER: 4
$B_p = 4$
key $= 3$
$R_p = 3$

1, 3, 5, 7, 9, 11

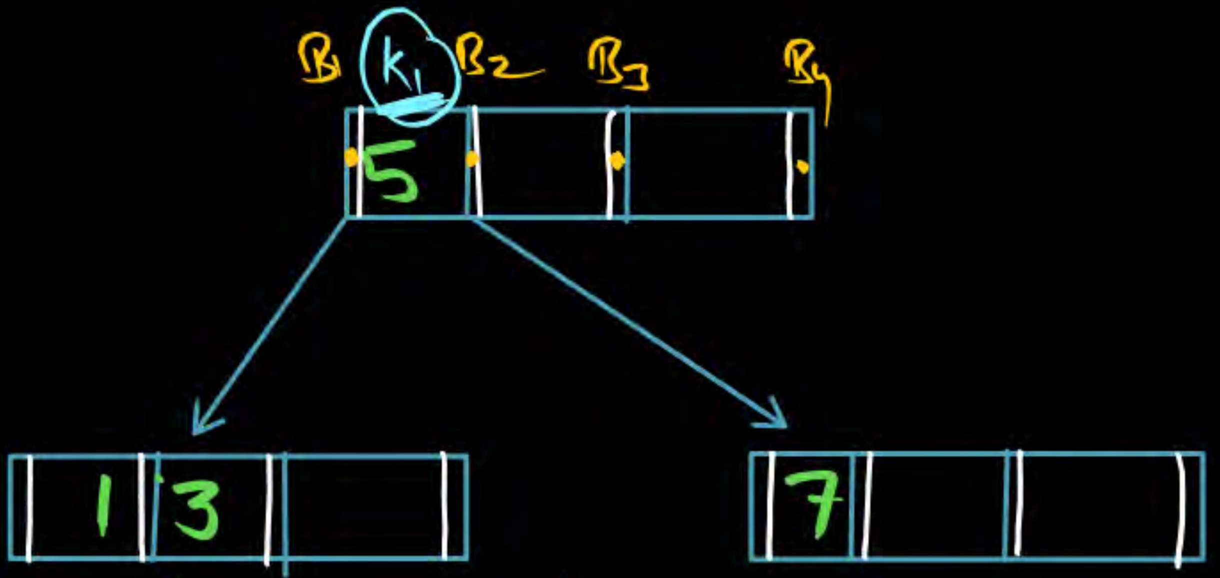


B Tree Defination violate



Right Biasing

OR



Left Biasing

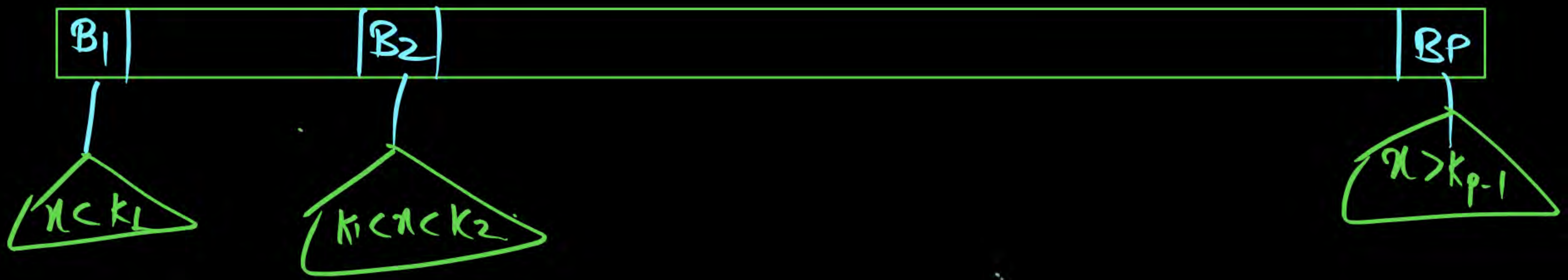
B Tree Definition

Balanced Tree

ORDER: P

① structure of Internal Node

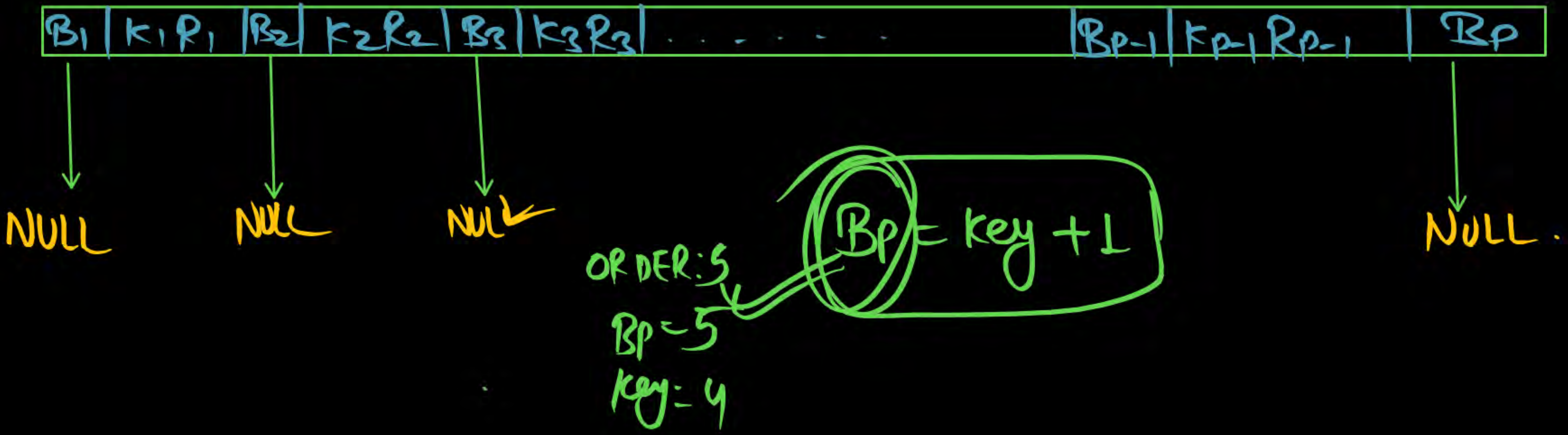
B_1	$K_1 R_1$	B_2	$K_2 R_2$	B_3	$K_3 R_3$	. . . .	B_{p-1}	$K_{p-1} R_{p-1}$	B_p



B Tree Definition

Balanced Tree

(2) structure of Leaf Node:



B Tree Definition

Balanced Tree

(3) Every Internal Node except the Root Node Contain at least $\lceil P/2 \rceil$ Block Pointer ($\lceil P/2 \rceil - 1$ keys) & atmost (P Block Pointer, (P-1) key)

(4) Root Can be Contain atleast 2 Block Pointer (1 Key) & atmost P Block Pointer (P-1 keys)

(5) All keys within the Node Must be assending order

(6) ALL the leaf Node should be at Same Level.

B Tree Definition

ORDER : P

Capacity of B Tree

B_1	$K_1 R_1$	B_2	$K_2 R_2$	B_3	$K_3 R_3$	. . . .	B_{P-1}	$K_{P-1} R_{P-1}$	B_P

$$\boxed{P * B_P + (P-1)\,\text{keys} + (P-1)\,R_P \le \text{BLOCK Size.}}$$

OR

$$\boxed{P * B_P + (P-1)[\text{Key} + R_P] \le \text{BLOCK size}}$$

P : order

B_P : Block | Index | Tree Pointer

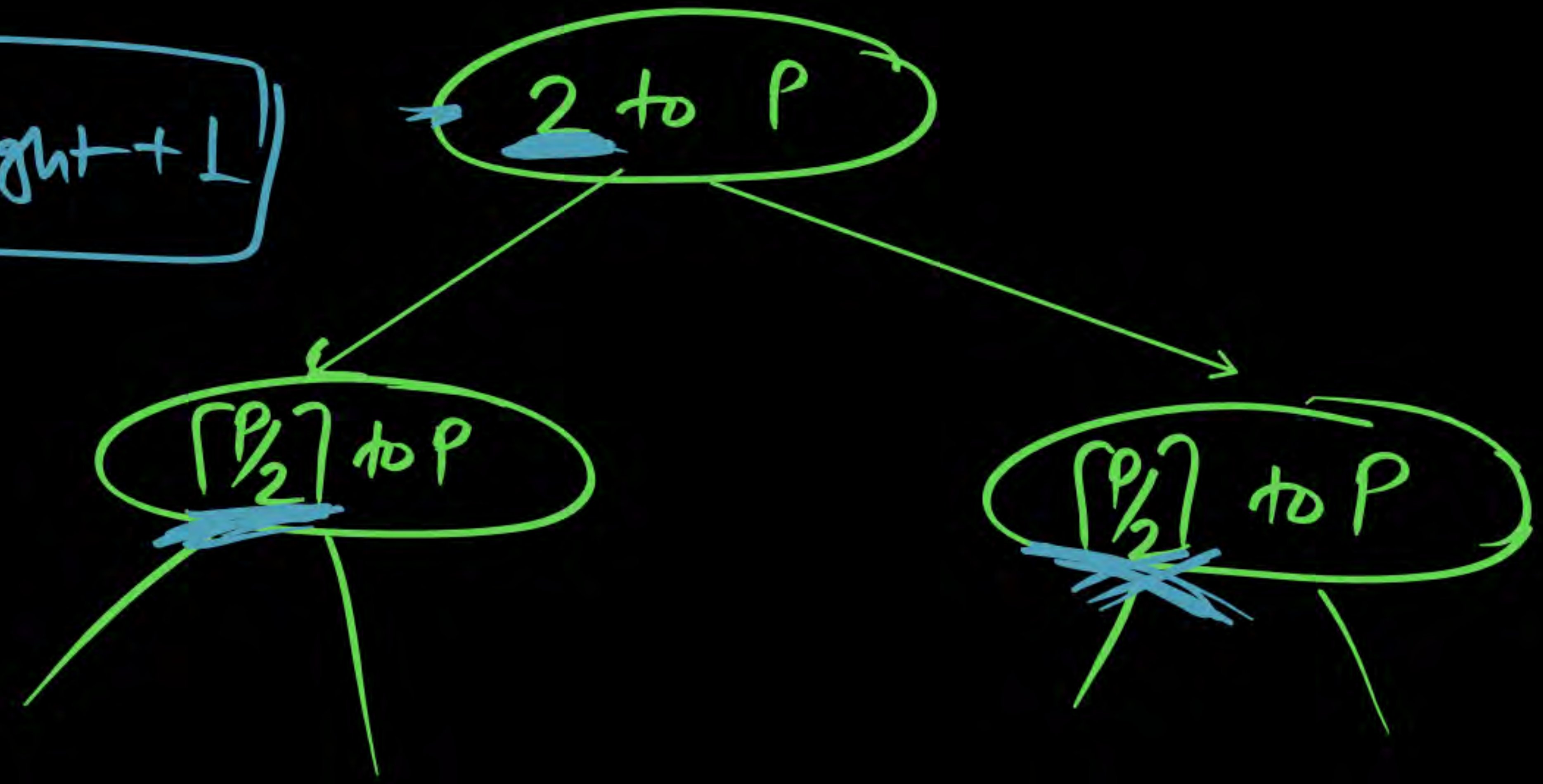
K : key | search key

R_P : Read | Data | Record Pointer.

B Tree Definition

Balanced Tree

$$\boxed{\text{Level} = \text{Height} + 1}$$



B Tree Definition

ORDER : 1

$p=5$

$\lceil p/2 \rceil = 3$

Height/Level	Min # Nodes	Min # BP	Min # Keys	
0/1	1	2	1	1
1/2	2	$2\lceil p/2 \rceil$	$2\left[\lceil p/2 \rceil - 1\right]$	$2(\lceil 5/2 \rceil - 1) = 4$
2/3	$2\lceil p/2 \rceil$	$2\lceil p/2 \rceil^2$	$2\lceil p/2 \rceil \left(\lceil p/2 \rceil - 1\right)$	$3 \times 4 = 12$
3/4	$2\lceil p/2 \rceil^2$	$2\lceil p/2 \rceil^3$	$2\lceil p/2 \rceil^2 \left(\lceil p/2 \rceil - 1\right)$	$3^2 \times 4 = 36$
4/5	$2\lceil p/2 \rceil^3$	$2\lceil p/2 \rceil^4$	$2\lceil p/2 \rceil^3 \left(\lceil p/2 \rceil - 1\right)$	$3^3 \times 4 = 108$
⋮				
h/h+1	$2\lceil p/2 \rceil^{h-1}$	$2\lceil p/2 \rceil^{h}$	$2\lceil p/2 \rceil^{h-1} \left(\lceil p/2 \rceil - 1\right)$	

161 Ans

$p = 5$

Q) ORDER : 5 $\underline{\text{Level : 5}} \rightarrow$ Height = 4.

Total Minimum # keys = ?

$\lceil p/2 \rceil = \lceil 5/2 \rceil = \lceil 2.5 \rceil = 3$

Ans) 161.

$2\left(\lceil p/2 \rceil - 1\right)$

$2\left(\lceil 5/2 \rceil - 1\right) \Rightarrow 2\,(3-1) \Rightarrow 2 \times 2$

$= 4$

B Tree Definition

ORDER : P.

$P = 5 \Rightarrow (P-1) = 4$

Height/Level.	maximum Max. # Nodes	Max. # Bp	maximum # Keys	
0/1	1	P	$(P-1)$	$4 = 4$
1/2	P	P^2	$P(P-1)$	$5 \times 4 = 20$
2/3	P^2	P^3	$P^2(P-1)$	$5^2 \times 4 = 100$
3/4	P^3	P^4	$P^3(P-1)$	$5^3 \times 4 = 500$
4/5	P^4	P^5	$P^4(P-1)$	$5^4 \times 4 = 2500$
				3124
h/h+1	P^h	P^{h+1}	$P^h(P-1)$	

B Tree Definition

Q Level 5, ORDER = 5

Total # Max keys = ?

B Tree Definition

Q.1 Maximum level ?

At each level Minimum # key & Min # BP.

$(\lceil P/2 \rceil - 1)$ $\lceil P/2 \rceil$

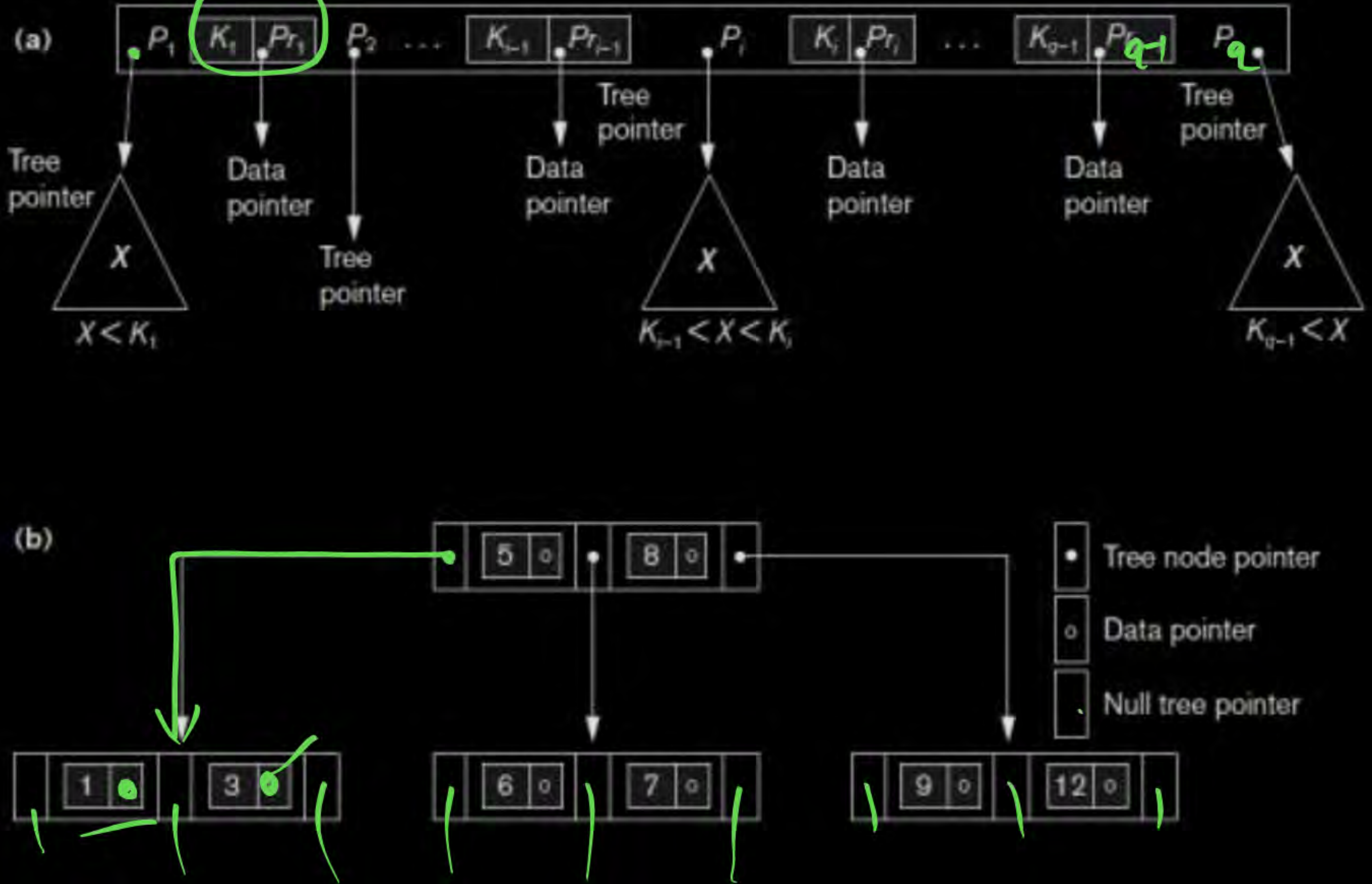
Q.2 Minimum level

At each level Maximum # Keys & Max # BP.

$[P-1]$ $[P]$

B Tree Structure

B-tree structures (a) A node in a B-tree with q−1 search values (b) A B-tree of order p=3. The values were inserted in the order 8, 5, 1, 7, 3, 12, 9, 6

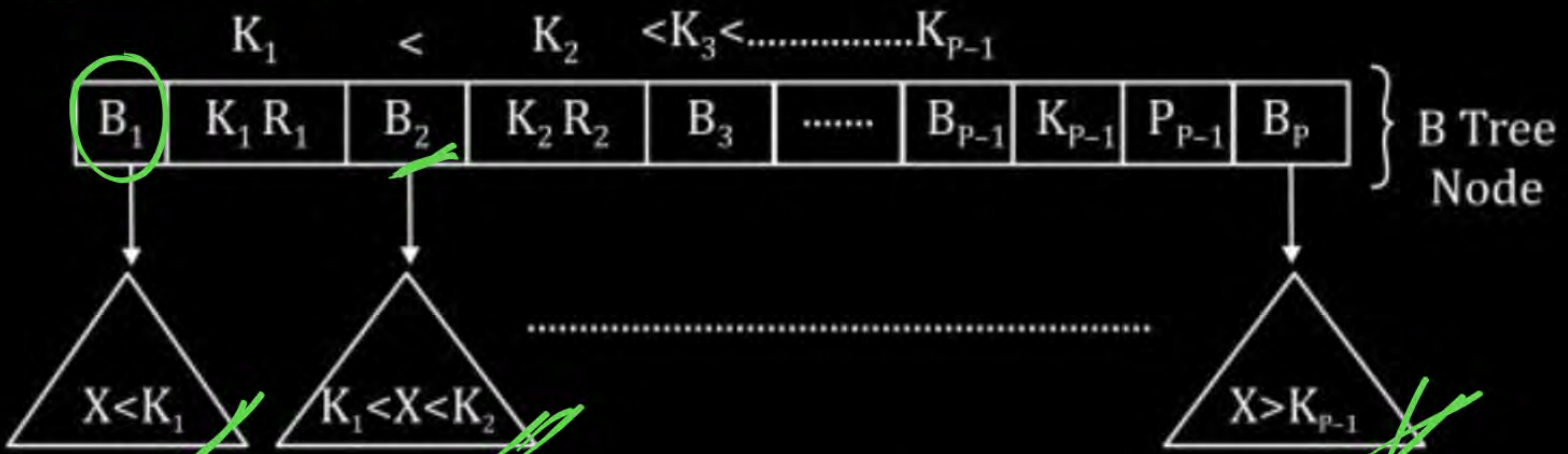


B Tree Definition

Order P: Max possible child pointers

[degree] can store in B Tree node.

(1) Node structure:



<P.Cp, (p-1)keys, (p-1)Rp>

(2) Every internal node except root must be at least ⌈P/2⌉ child pointer with (⌈P/2⌉ − 1) key's and at most P children pointer with (P - 1) key's must.

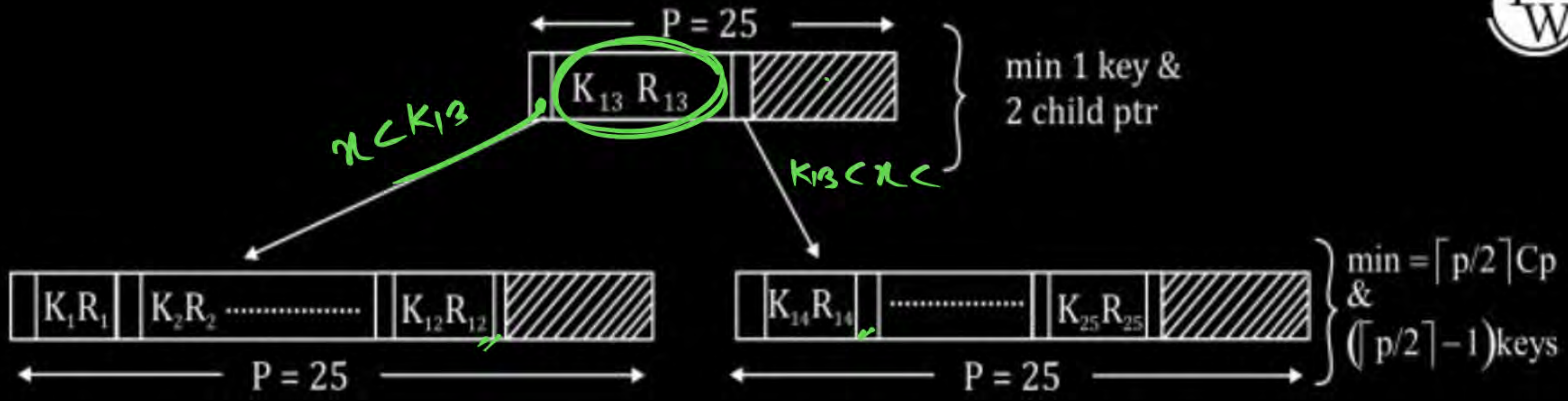
(3) Root node can be at least 2 children with 1 key, at most P children pointer and (P - 1) key's.

(4) Every leaf node must be at same level and keys with in node should be sequence order (Increasing order)

Order P: 25 [Max Cp in node]

⇓ Node Splitting

$$\underbrace{K_1 < K_2 < K_3 \ldots\ldots}_{\text{left Subtree}} < K_3 < \underbrace{K_{14} \ldots\ldots K_{25}}_{\text{Right Subtree}}$$



except Root Every
index node
atleast 50% occupied
in B/B+ Tree index

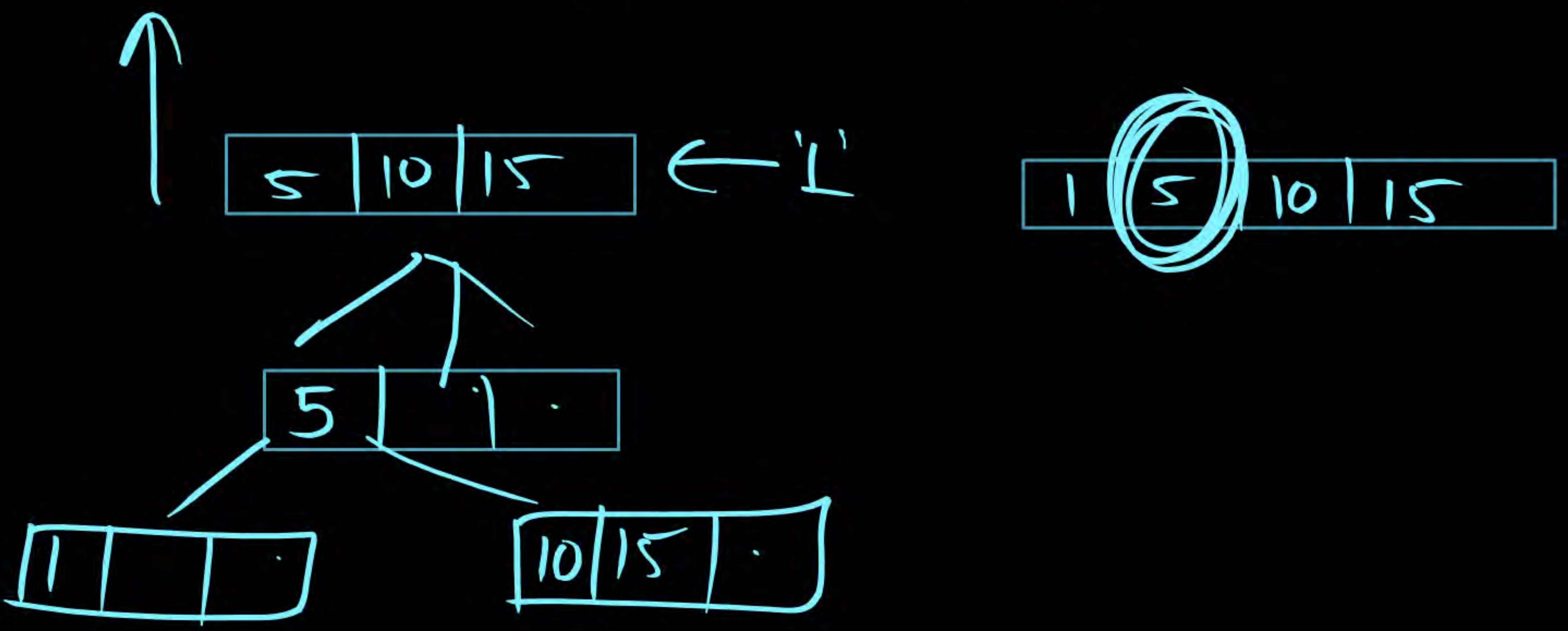
Block B tree node
[Free]

Q.

Construct B Tree with order P: 4[max Bp/cp (Block/child pointer) per node] and following sequence of key's ____________

5, 10, 15, 1, 23, 34, 13, 8.

P=4   key = 3



Q.3 Consider a table T in a relational database with a key field K. A B-tree of order p is used as an access structure on K, where p denotes the maximum number of tree pointers in B-tree index node. Assume that K is 10 bytes long; disk block size is 512 bytes; each data pointer P_D is 8 bytes long and each block pointer P_B is 5 bytes long. In order for each B-tree node to fit in a single disk block, the maximum value of p is [GATE-2004 : 2 Marks]

(A) 20

(B) 22

(C) 23

(D) 32

key = 10 Byte

Rp = 8 Byte

Bp = 5 Byte

Block Size = 512 Byte

$$P \times B_p + (P-1)\,key + (P-1)\,R_p \leq Block\ Size$$

$$P \times 5 + (P-1)10 + (P-1)8 \leq 512$$

$$5P + 10P - 10 + 8P - 8 \leq 512$$

$$23P - 18 \leq 512$$

$$23P \leq 530$$

$$P = \left\lfloor \frac{530}{23} \right\rfloor = \lfloor 23.04 \rfloor = 23 \text{ Ans}$$

if we taken 24, then its exceeds by Block size [512 Byte].

Advantage:

1) B Tree index best suitable for Random access of some record.

select *

FROM R

WHERE A = 24 ;

One record access

I/O cost: K + 1 blocks

$\approx [\log_p n] + 1 = \theta(\log_p n)$

Disadvantage:

1) B Tree index not best suitable for sequency access of range of records.

select *

FROM R

WHERE $\boxed{A \geq 30 \text{ and } A \leq 85;}$ → Range of record Access required

← X blocks of DB →

$\underbrace{\text{I/O cost: } x[\log_p n + 1] + \text{cost of unsuccessful}}_{\text{More Access cost}}$

More Access cost

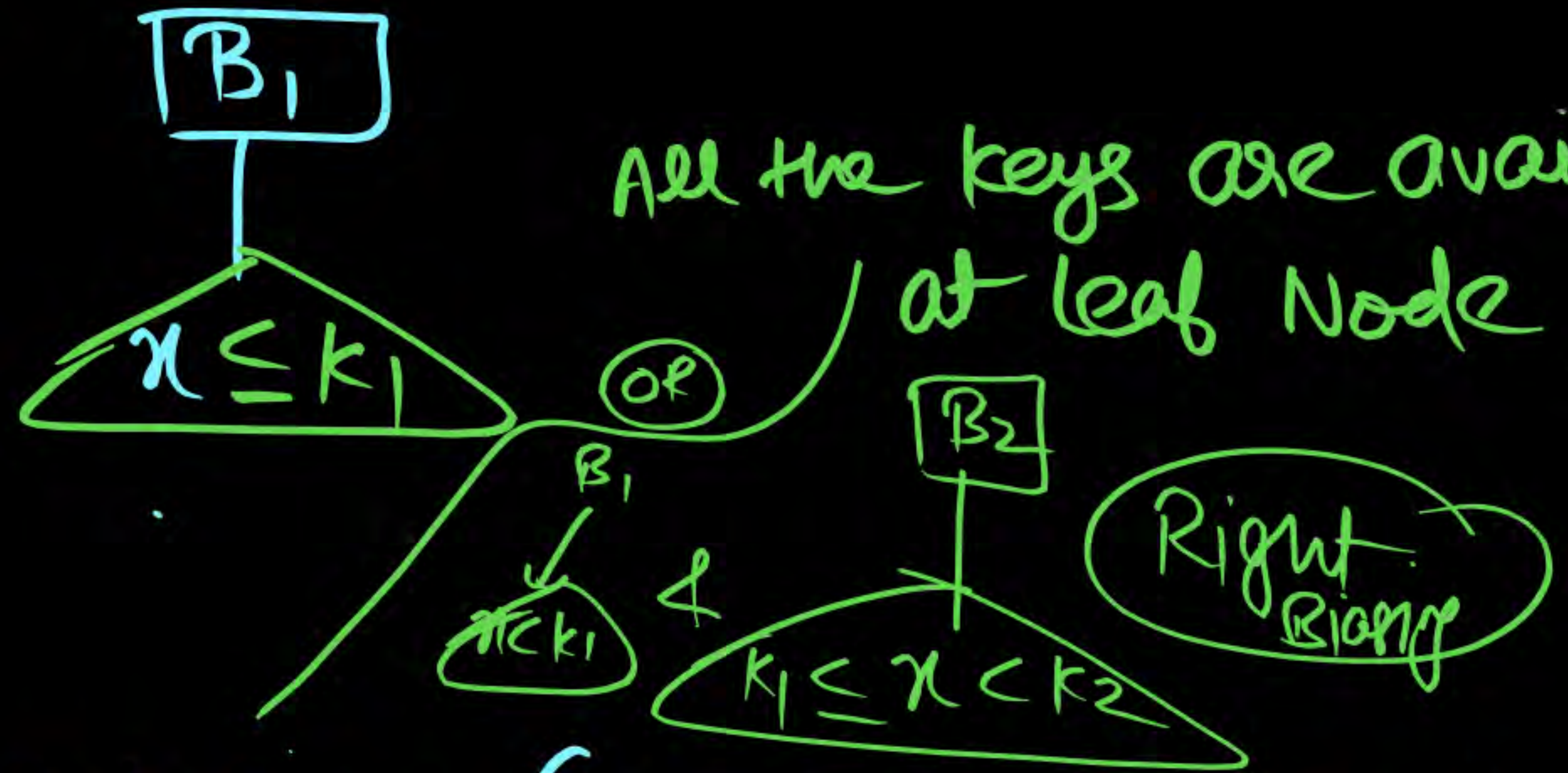
[Unordered File DB]

B^+ Tree Definition

Total 3 Difference

① In Internal Node ⇛ No Read (Record) Pointer.

②



All the keys are available at leaf Node.

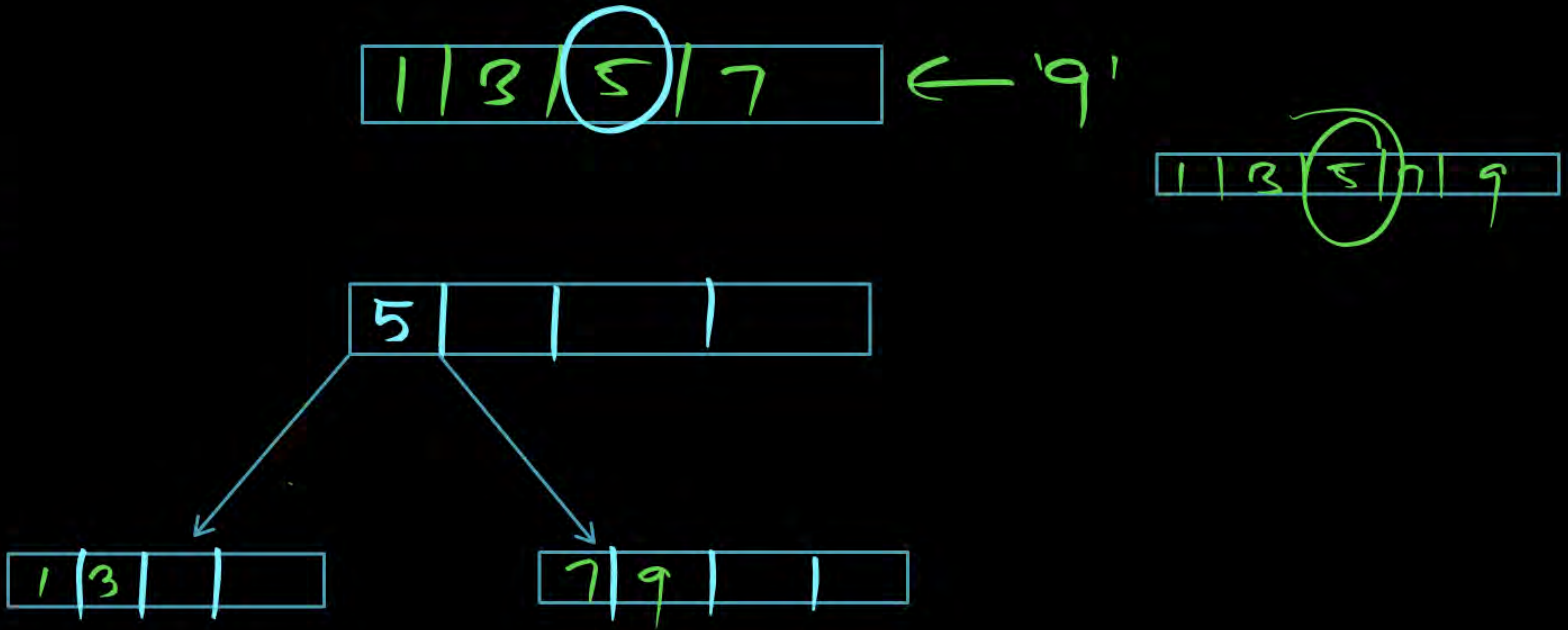
③ Leaf Node Contain (key + Rp) Pair + 1 Block Pointer.

B⁺ Tree Definition

ORDER : 5

1, 3, 5, 7, 9, 11

BTree

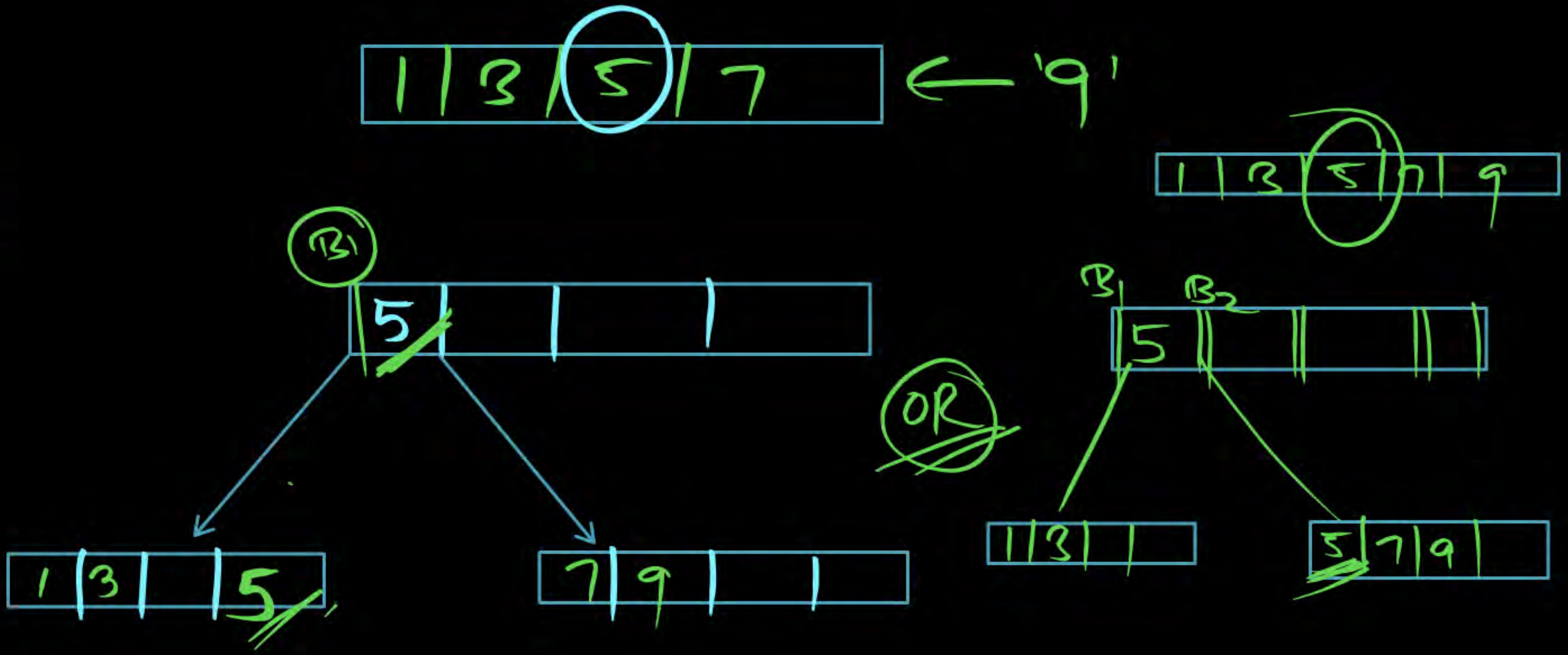


B+ Tree Definition

ORDER : 5

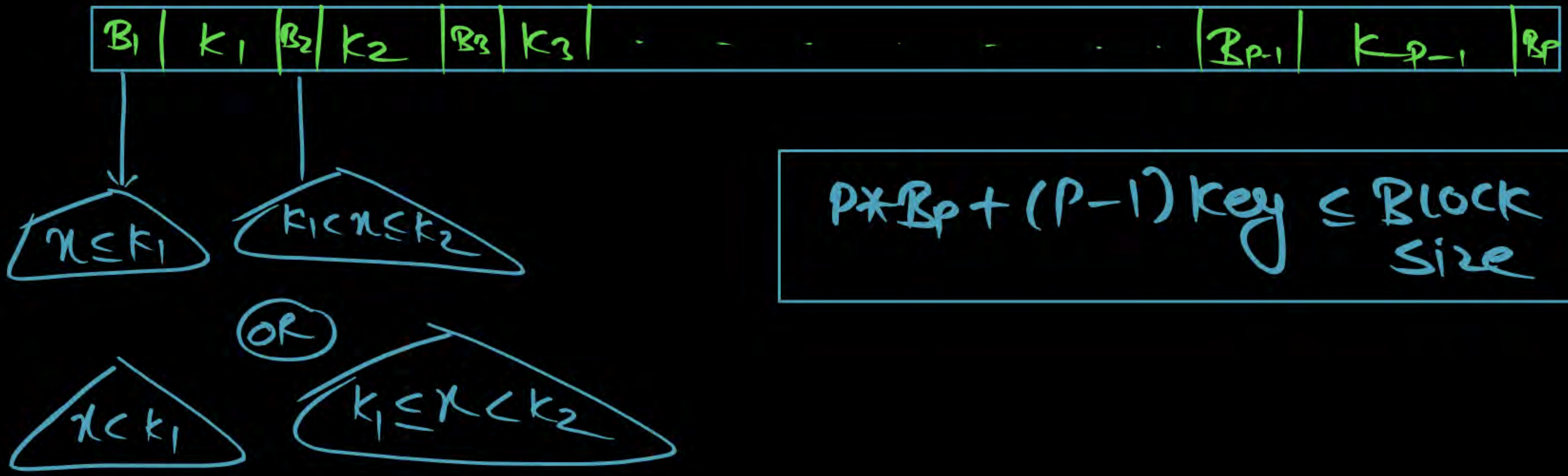
1, 3, 5, 7, 9, 11

B+ Tree



B+ Tree Definition

① Structure of Internal Node



$$P * B_p + (P-1)\,\text{Key} \le \text{Block size}$$

B^+ Tree Definition

② Structure of leaf Node

$K_1 R_1$	$K_2 R_2$	$K_3 R_3$	. . . .	$K_{P-1} R_{P-1}$	. .	$\bullet \rightarrow$

Block Pointer

$(\text{Key} + R_P) + 1 \text{ Block Pointer}$

Leaf Node.

$$\boxed{(P-1)\text{keys} + (P-1)R_P + 1 B_P \leq \text{Block size.}}$$

B^+ Tree Definition

③ Internal Node min $\lceil P/2 \rceil$ Bp & max P Bp

④ Root Node min 2 Bp & max P Bp.

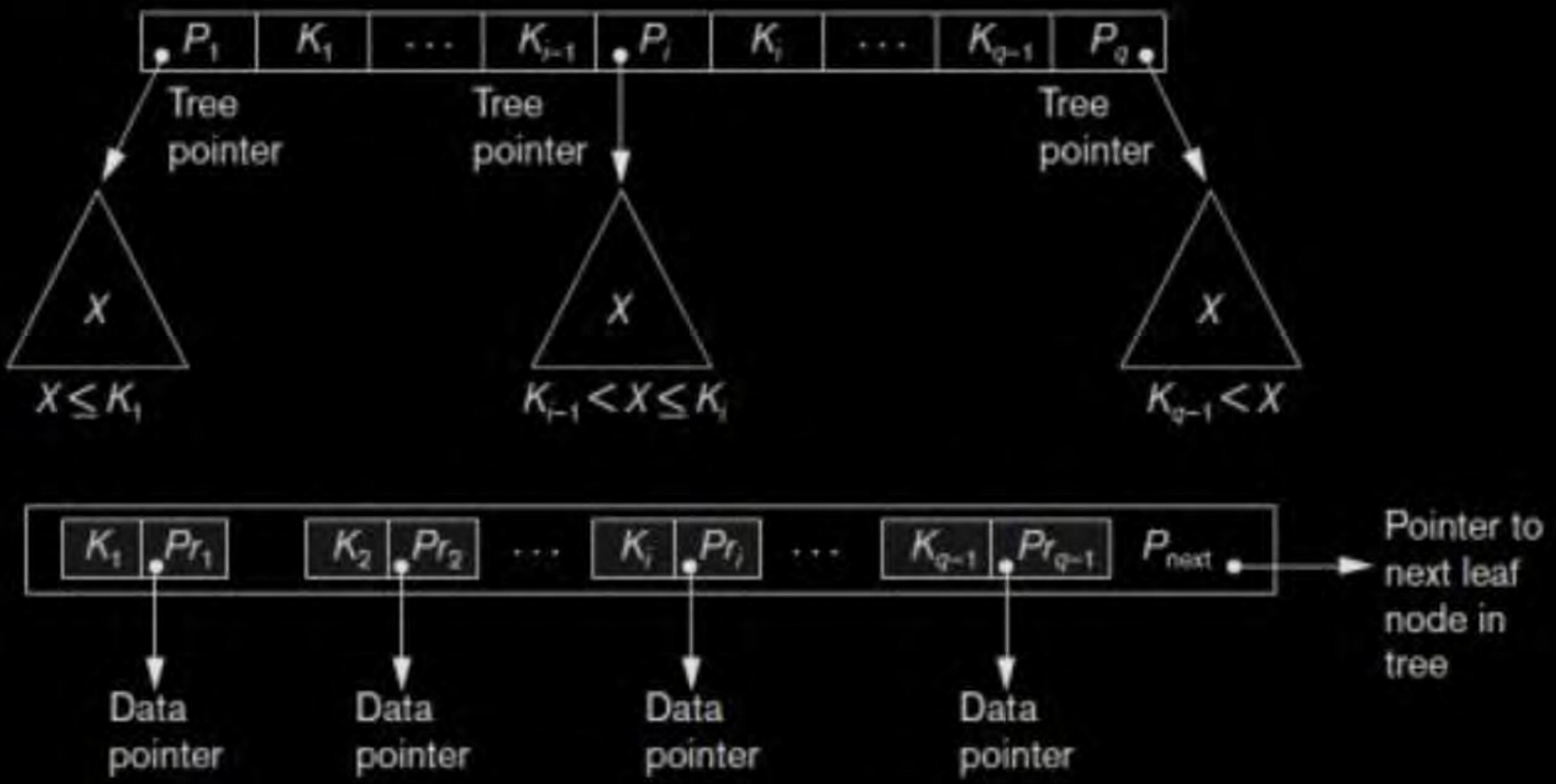
⑤ All the Leaf Node Should be at Same Level.

⑥ Keys within the Node Should be in assending order.

B+ Tree

(a)

(b)



The nodes of a B+-tree (a) Internal node of a B+-tree with q−1 search values (b) Leaf node of a B+-tree with q−1 search values and q−1 data pointers

Non Leaf | Internal Node

$$\boxed{P * B_P + (P-1)\,\text{keys} < \text{BLOCK size}}$$

B^+ Tree Definition

Order P: max possible pointers [degree] can store in B^+ Tree node.

(1) Node Structure:

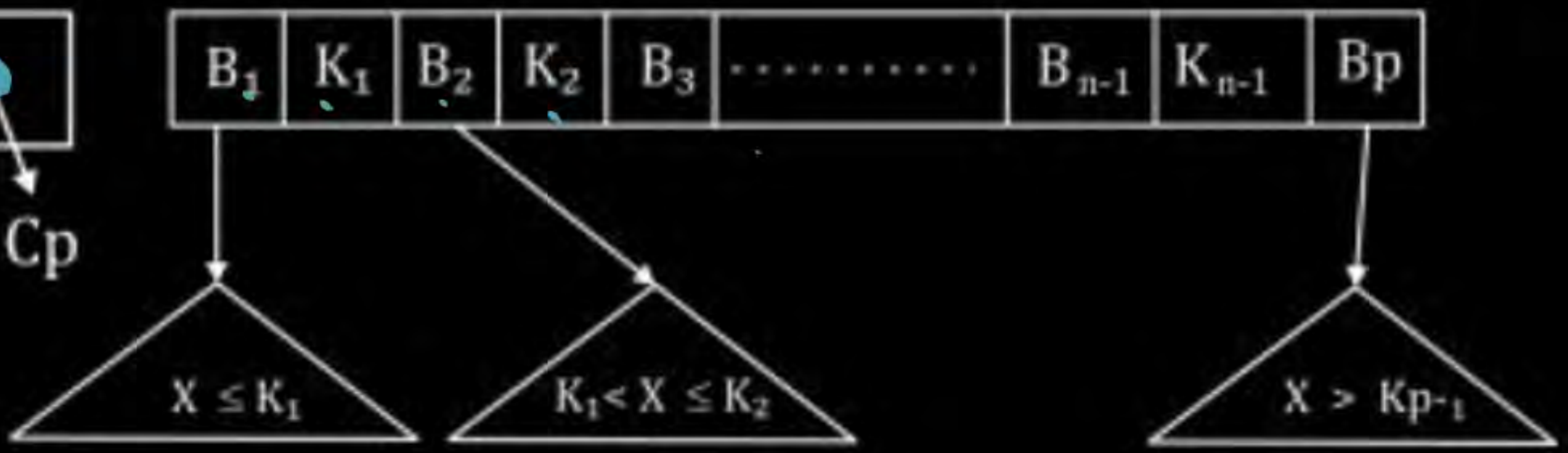
(a) Leaf Node

[set of (key, Rp) pair and only one block pointer pointed to next Leaf node]

(b) Internal Node

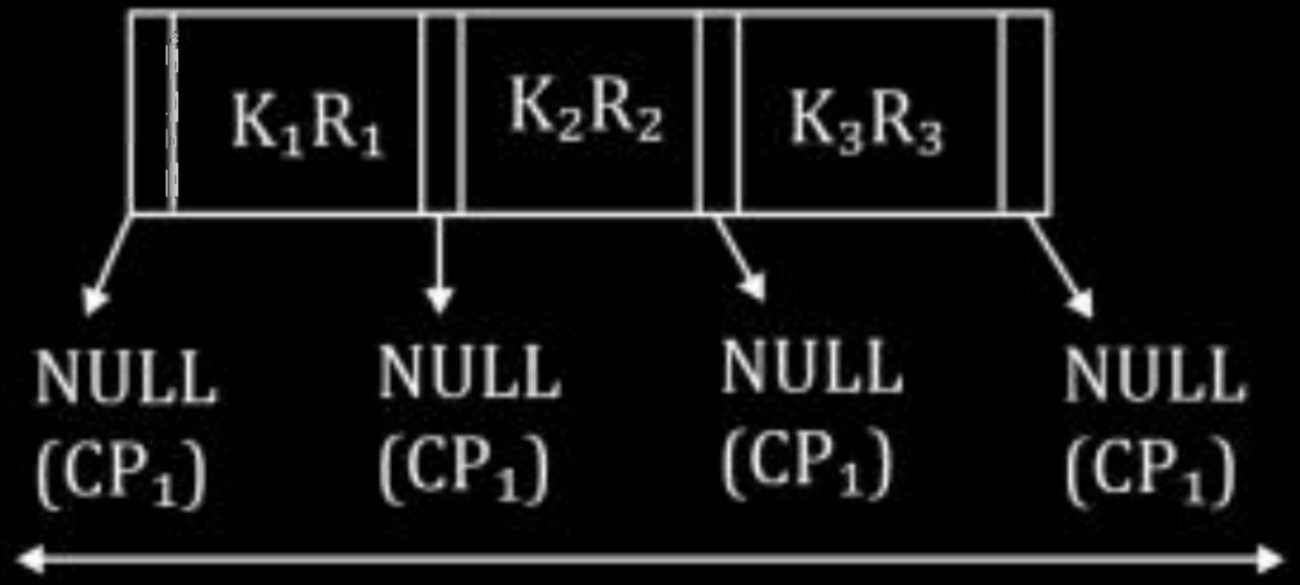
[key's and child pointer]

∴ No record pointer (Rp).

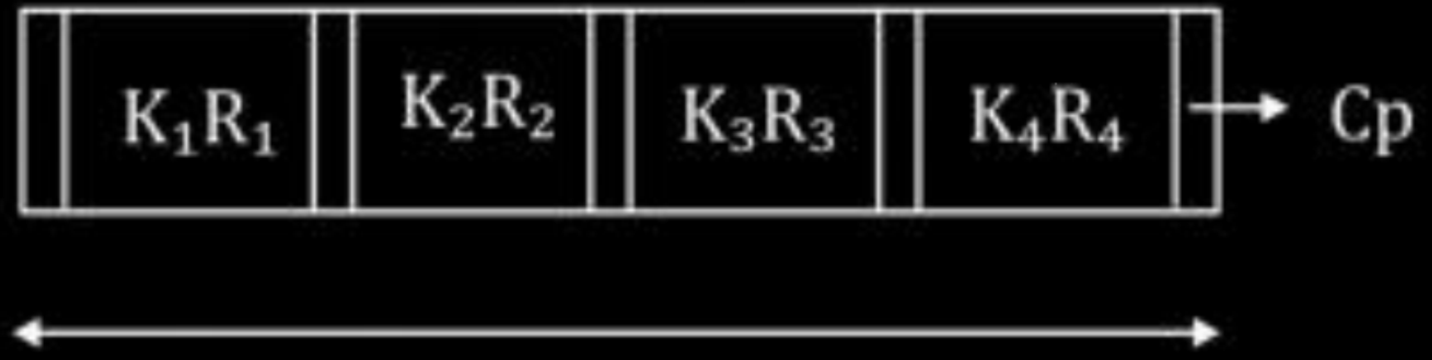


⇒ Left Biasing condition $\boxed{k_1 < x \leq k_2}$ ....

Right Biasing condition $\boxed{k_1 \leq x < k_2}$ .... $\boxed{x \geq k_{p-1}}$



B Tree Leaf



B^+ Tree Leaf

⇒ Remaining Balancing Condition same as B Tree

Advantage:

1) B^+ Tree index best suitable for sequence access of range of records.

   [Range Quires runs faster using B^+ Tree index]

   Select *

   FROM R

   WHERE $A \geq 30$ and $A \leq 85$;

   I/O cost: $[\log_p n + \underbrace{x}_{\text{For Index}} + \underbrace{x}_{\text{For Database File}}]$

Q.4 The order of an internal node in a B^+tree index is the maximum number of children it can have. Suppose that a child pointer takes 6 bytes, the search field value takes 14 bytes, and the block size is 512 bytes. What is the order of the internal node?

(A) 24

(B) 25

(C) 26 ✓

(D) 27

$\boxed{P * B_P + (P-1)\text{keys} \le \text{Block size}}$

$6P + (P-1)14 \le 512$

$20P - 14 \le 512$

$20P \le 526$

$P = \frac{526}{20} = \lfloor 26.3 \rfloor = 26$

$B_P = 6$

$\text{key} = 14$

Q.5 Consider a B^+-tree in which the maximum number of keys in a node is 5 . What is the minimum number of keys in any non-root node?

(A) 1

(B) 2

(C) 3

(D) 4

Maximum # keys $(P-1) = 5$

$P-1=5$

$\boxed{P=6}$ ORDER $=6$

Minimum # keys $= \lceil P/2 \rceil - 1 \Rightarrow \lceil 6/2 \rceil - 1 = 3-1 = 2$

JOIN with foreign key Concept.

Note

Whenever Two Tables are Joined (Natural Join) with Respect to Primary key & Foreign key then maximum Number of Tuples in the Resulting Relation is equal to Tuples in the Referencing [containing F.K] Relation.

Q.1

Consider the relations:

F.K

Supply (<u>SupplierID, Itemcode</u>) with 1000 tuples

Inventory (<u>Itemcode</u>, color) with 2500 tuples

Let p and q be the number of maximum and minimum records in Supply JOIN Inventory, the value of p + q is ________. (Itemcode is FK in Supply table).

Refering F.K

Supply (<u>SupplierID, Item Code</u>) Inventory (<u>Item Code</u>, Color)

1000 Tuple 2500 Tuple

Maximum # Tuples in JOIN [p] = 1000 Ans

Minimum # Tuple [q] = 1000

$p + q = 1000 + 1000$

$= 2000$ Ans

The value Present in the foreign key Must be Present in Primary key

Foreign key May Contain Dublicates & NULL value

Here Item Code is Foreign key in Supply Table Reference P.K of Inventory table

But Here Item Code is also key in Supply Table

SUPPLY

Sid	Icode
S1	C1
S1	C2
S2	C1
S2	C2

IC
C1
C2
C3
C4
C5
C6

SUPPLY (Supplier id Item Code)

Here So Item Code Cannot NULL.

Q. 2

Consider the relations:

Employee(<u>Eid</u>, Ename) with 2000 tuples

Department(<u>Eid, Did</u>) with 1000 tuples

Let p and q be the number of maximum and minimum records in Employee JOIN Department, the value of p + q is ________. (Eid is FK in Department table).

2000

Q.3

Consider the relations:

R1($\underline{A}$, B, C) with n tuples

R2($\underline{B}$, D, E) with m tuples (B is FK in R1).

maximum tuples in $R1 \bowtie R2$ = n

minimum tuples in $R1 \bowtie R2$ =

Referencing

$R_1(\underline{A}\ B\ C)$ $\xrightarrow{F.K}$ $R_2(\underline{B}\ D\ E)$

n Tuple   m Tuple

Maximum $R_1 \bowtie R_2 = n$

Minimum $R_1 \bowtie R_2 = n$ (if foreign key Reference to PK & F.K Not NULL)

$= 0$ [Not a foreign key OR NULL]

Q.4

Consider the relations:

R1(<u>A</u>, B,C) with n tuples

R2(<u>B</u>, D,E) with m tuples (B is Not FK in R1).

maximum tuples in $R1 \bowtie R2$

minimum tuples in $R1 \bowtie R2$

Q.5 Consider the join R ⋈ S between R (A B C) and S (A D E) with attribute A being the primary key in both relations and attribute A in S is a Foreign Key referencing to attribute A in R. R has m tuples and S has n tuples, maximum and minimum number of tuples in R ⋈ S respectively are?

(a) m, m + n

(b) m*n, 0

(c) min (m, n), min (m, n)

(d) max (m, n), min (m, n)

R(ABC

A	B	C
1		
2		
3		
4		
5		
6		

S(ADE)

A	D	E
1		
2		
3		
4		

Q. 6

Consider the relations:

R1(A, B, C) with n tuples

R2(D, B, E) with m tuples

$\{A \to B,\ B \to C,\ D \to B,\ B \to E\}$

maximum tuples in $R1 \bowtie R2$

minimum tuples in $R1 \bowtie R2$

$R_1(ABC)$ $R_2(DBE)$

B is Not key

C.K : A C.K : D

Maximum $R_1 \bowtie R_2 = n * m$

Minimum $= 0$

[B is Non key Attribute in Both Table]

MCQ

The following functional dependencies hold for relations R(A, B, C) and S(B, D, E) FD for both the tables.

$B \rightarrow A$

$A \rightarrow C$

The relation R contains 200 tuples and the relation S contains 100 tuples. What is the maximum number of tuples possible in the natural join $R \bowtie S$?

[GATE-2010-CS: 2M]

(A) 100

(B) 200

(C) 300

(D) 2000

THANK YOU!