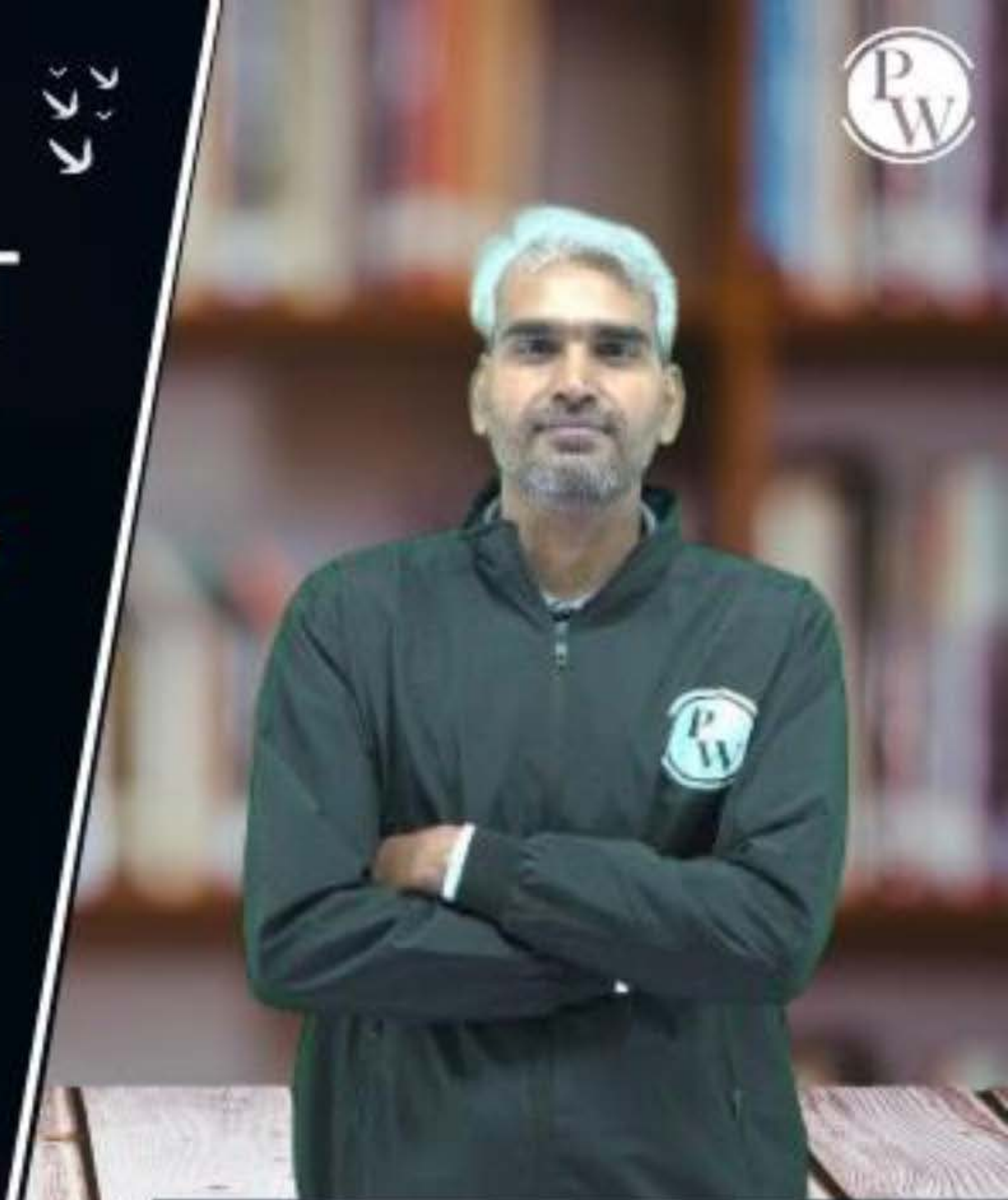
Computer Science & IT

Database Management System

File organization and indexing

Lecture No. 04



By- Vishal Sir

Recap of Previous Lecture

Topic — Multi-level index tree

Topic — Structure of B tree

Slide

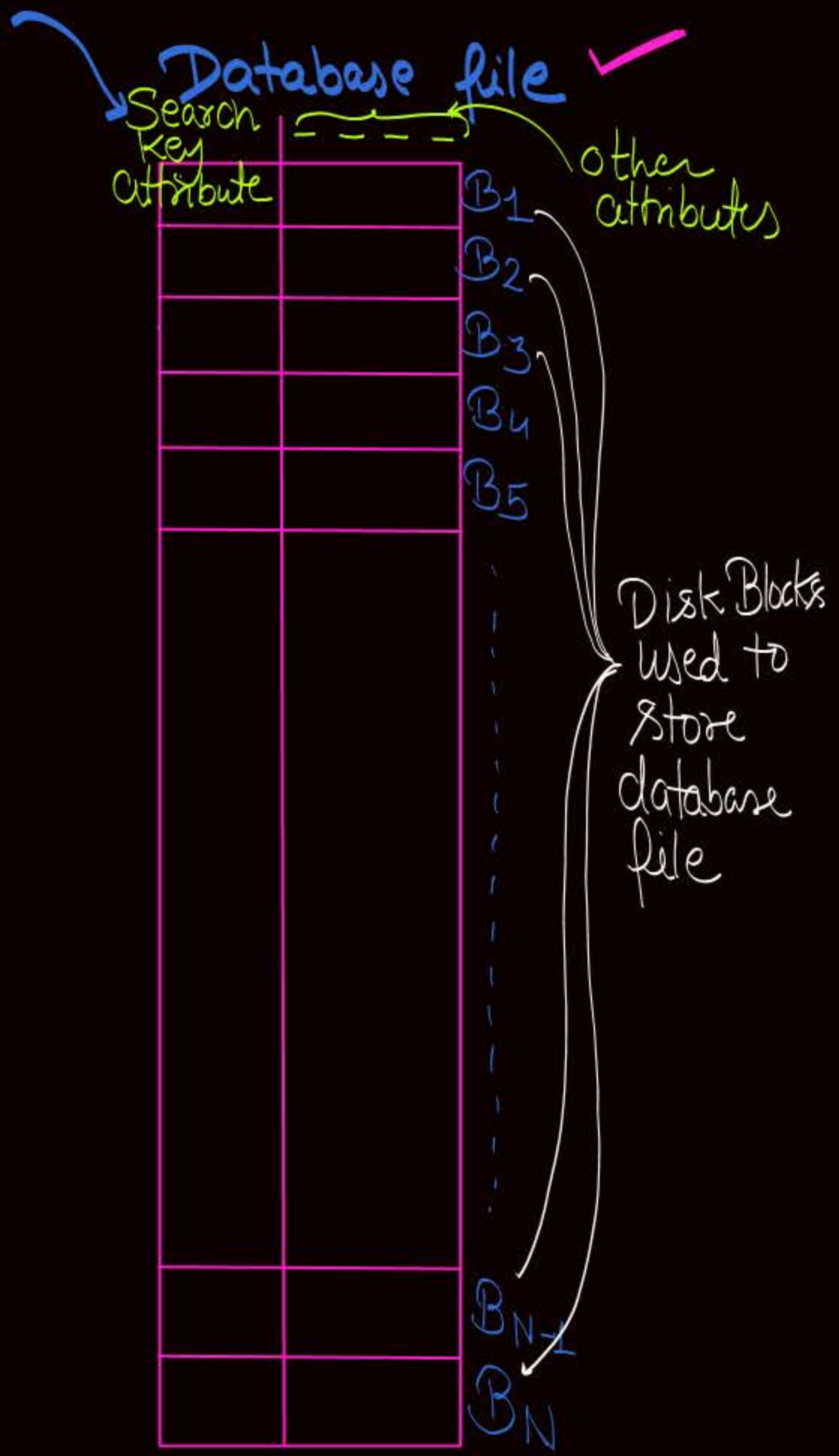
Topics to be Covered

- **Topic** Structure of B+ tree
- **Topic** Insertion into B tree
- **Topic** Deletion from B tree

Slide

Topic : Multi-level Index

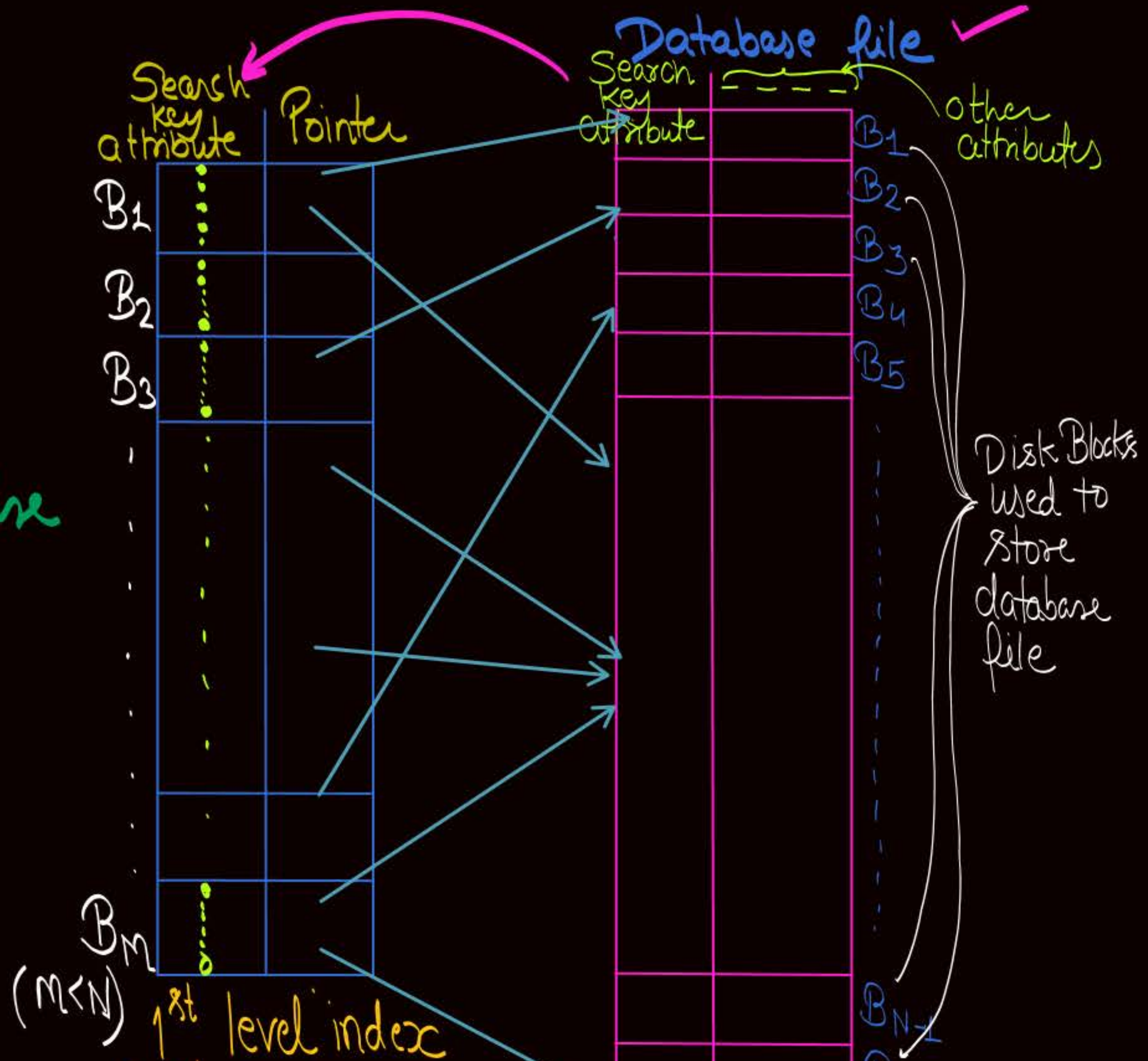
Multi-level index is used to reduce the IO Cost w.r.t. single level index



Database file
Search key attribute
other attributes
B1
B2
B3
B4
B5
BN-1
BN
Disk Blocks used to store database file

$\lceil \log_2 M \rceil + 1$

Will be Worst Case IO Cost using single level index

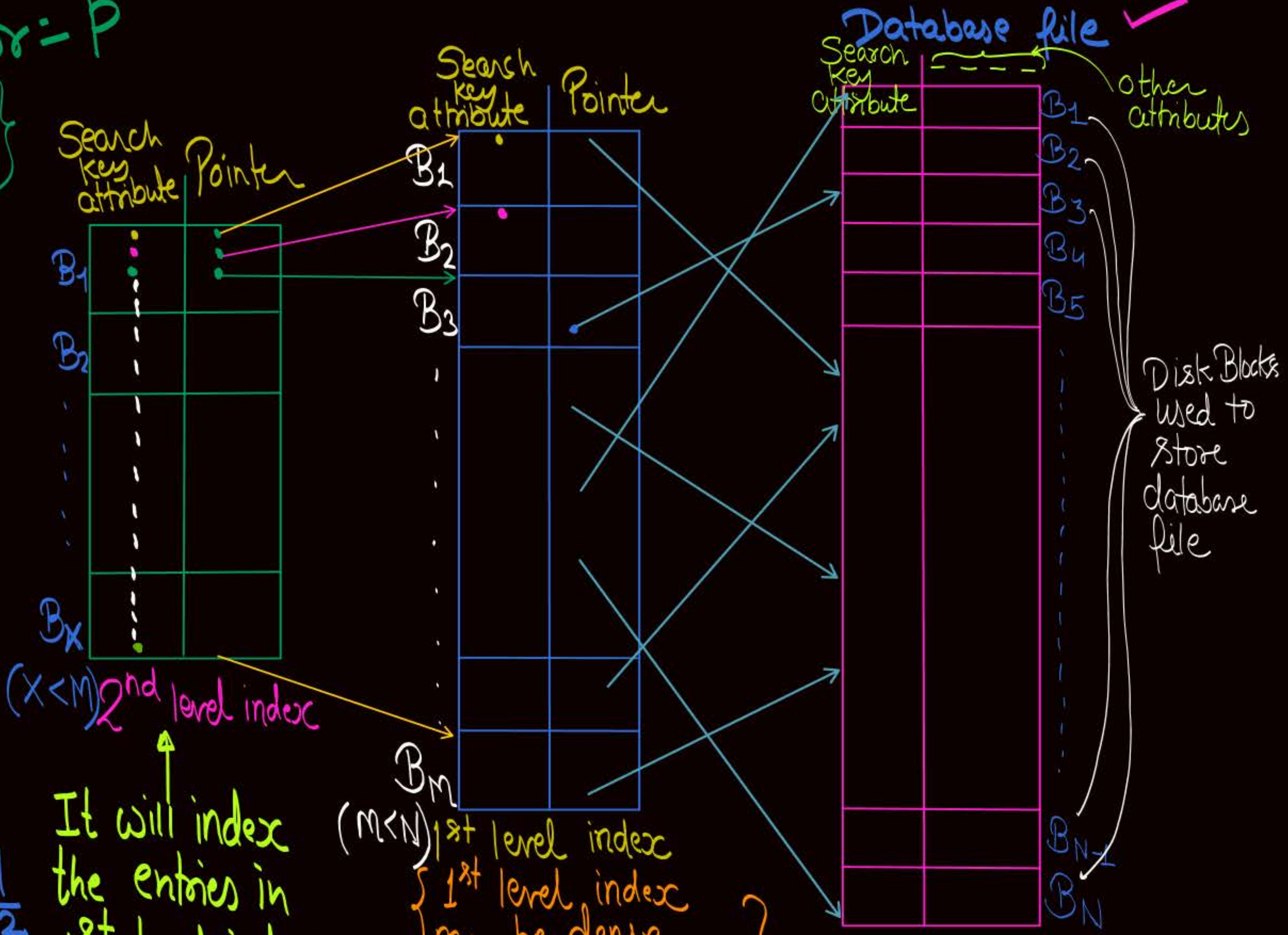


{1^{st} level index
May be dense or May be Sparse}

let Blocking factor = P
{w.r.t. blocks of 2nd level index}

then

$$\#\text{ disk blocks required for 2nd level index} = \left\lceil \frac{M}{P} \right\rceil$$



{1st level index may be dense or may be sparse}

$$\#\text{ Disk block required for 3rd level index} = \left\lceil \frac{\left\lceil \frac{M}{P} \right\rceil}{P} \right\rceil \approx \frac{M}{P^2}$$

It will index the entries in 1st level index
(it will be sparse index)

(Key size + Pointer size)

Entry size : 50B

leaf → Previous → Previous → Previous - - -

$M \quad \frac{M}{2} \quad \frac{M}{2^2} \quad \frac{M}{2^3}$

$\left\lceil \log_{P \approx 30} M \right\rceil$

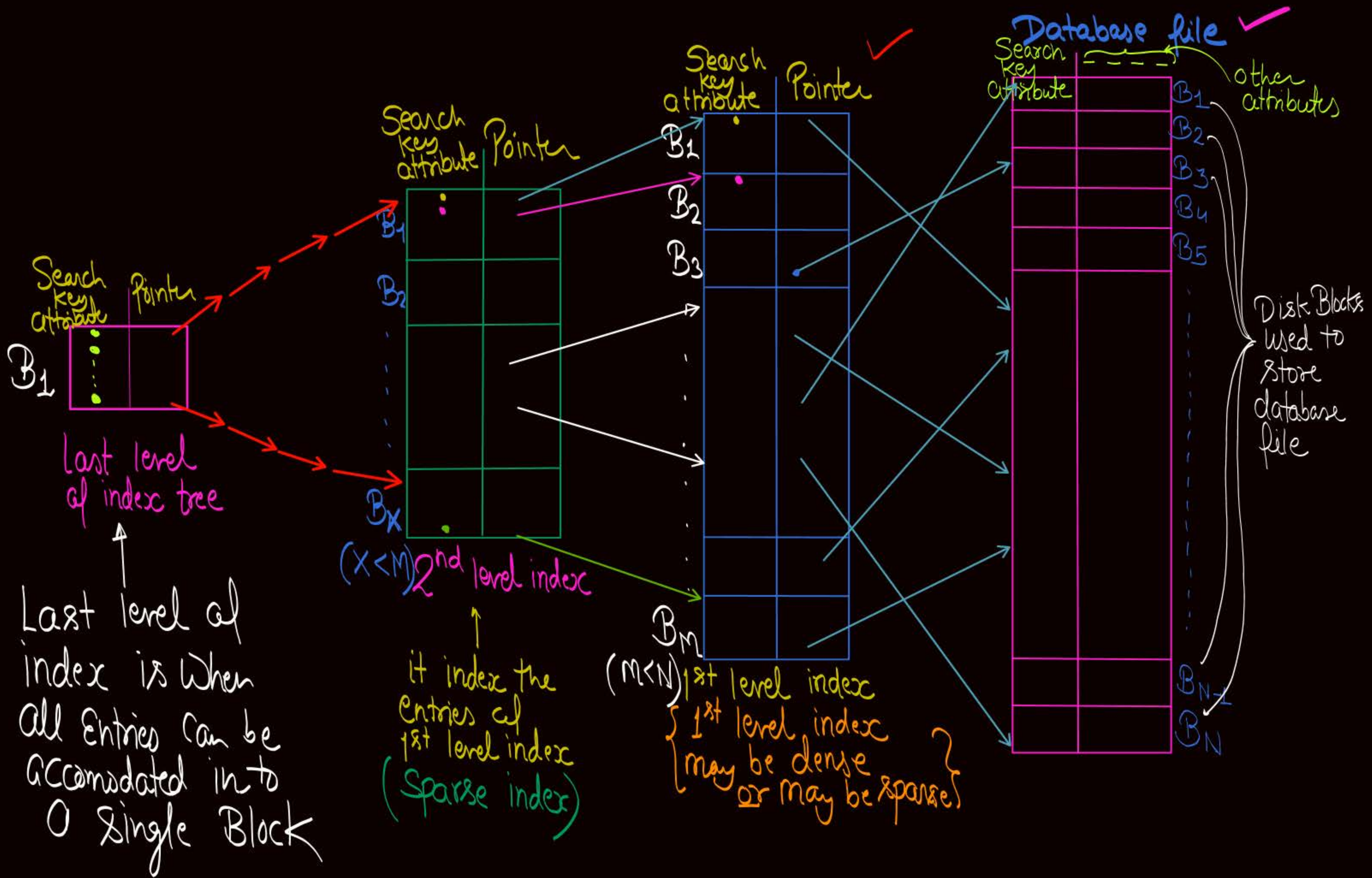
$$\boxed{\log_2 M \gg \log_{30} M}$$

$$Bf = \frac{1\,KB = \text{Block size}}{50\,B = \text{Entry size}}$$

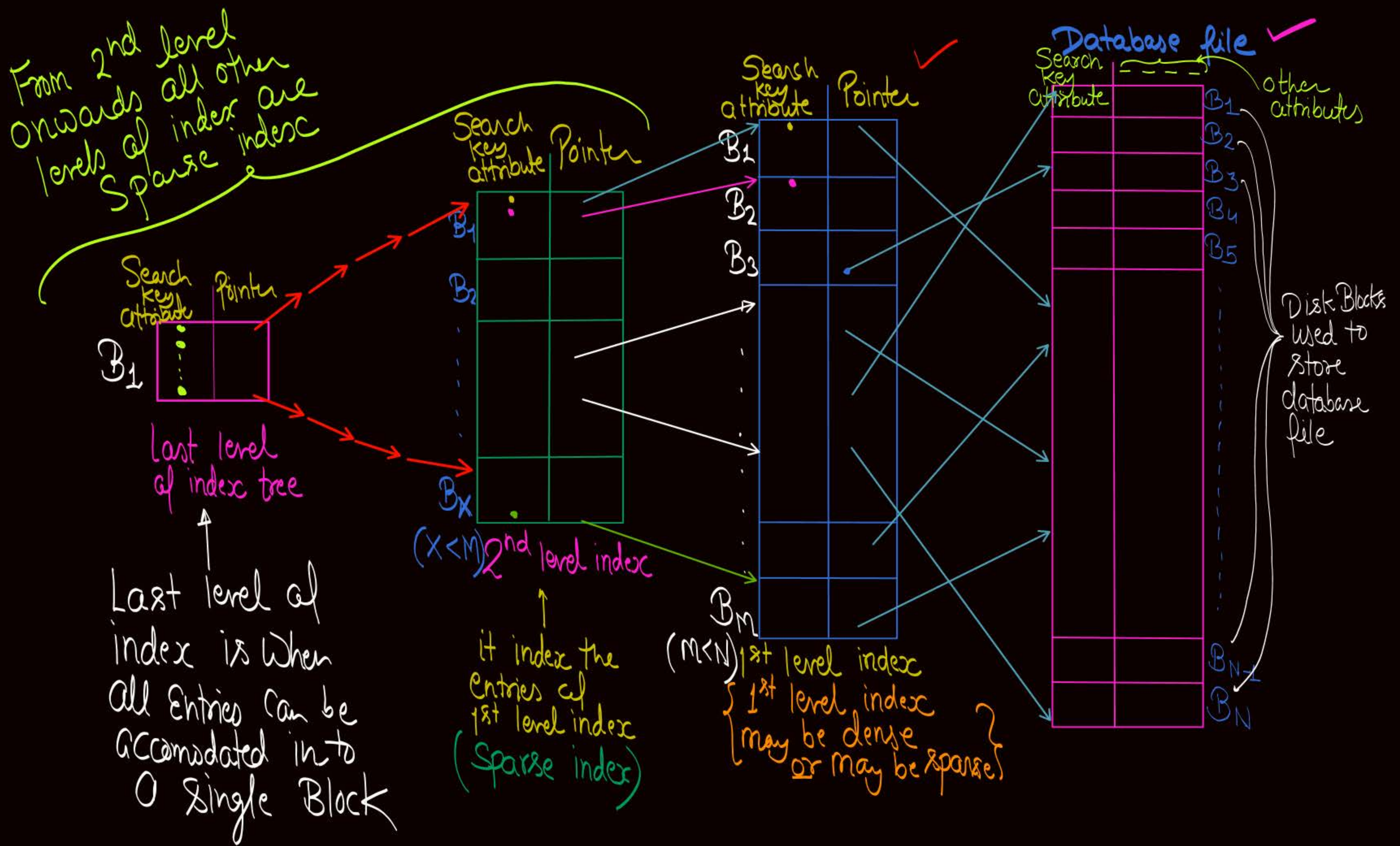
$Bf \approx 20$

$P \quad P^2 \quad P^3$

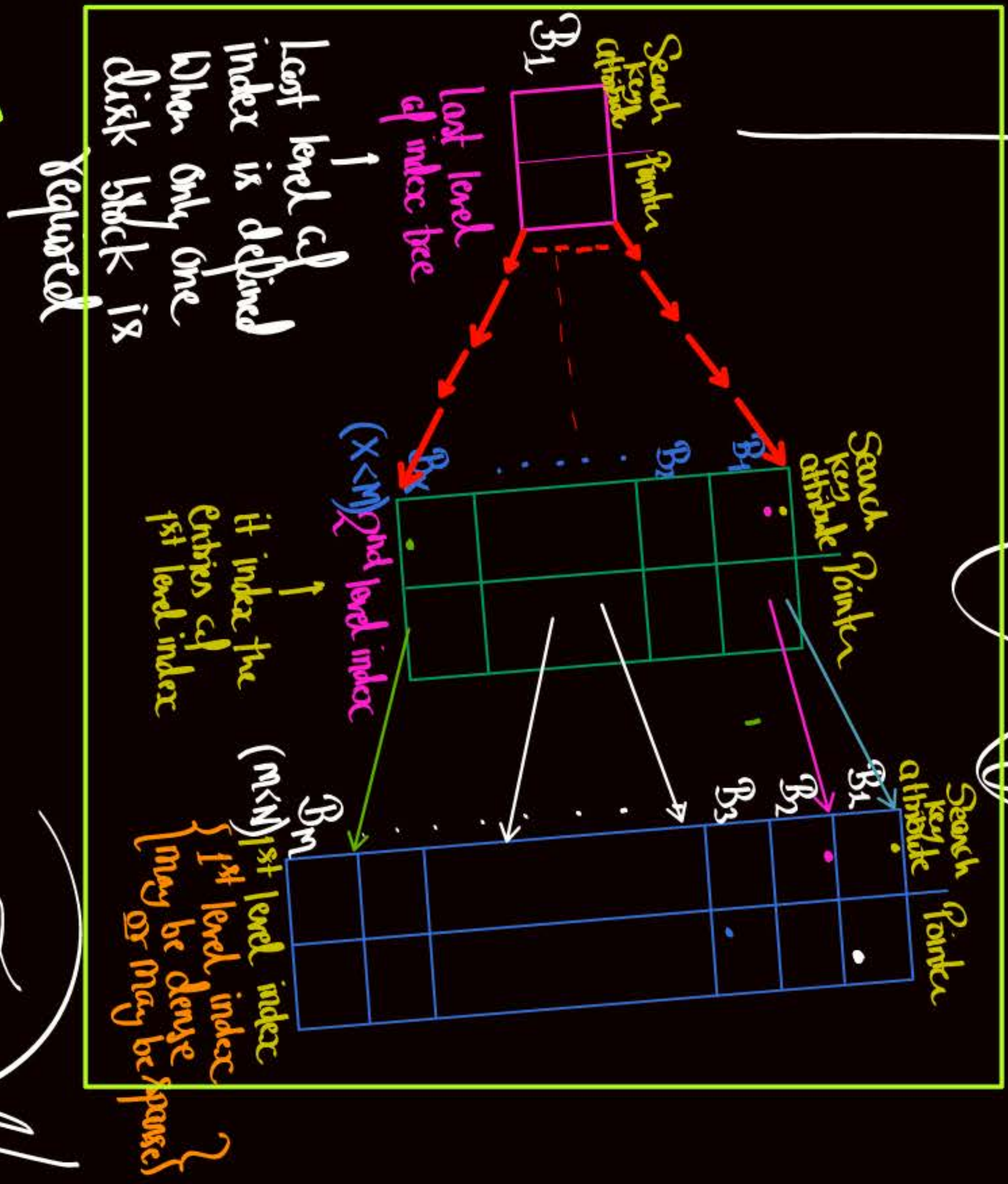
$20^2 \quad 20^3$

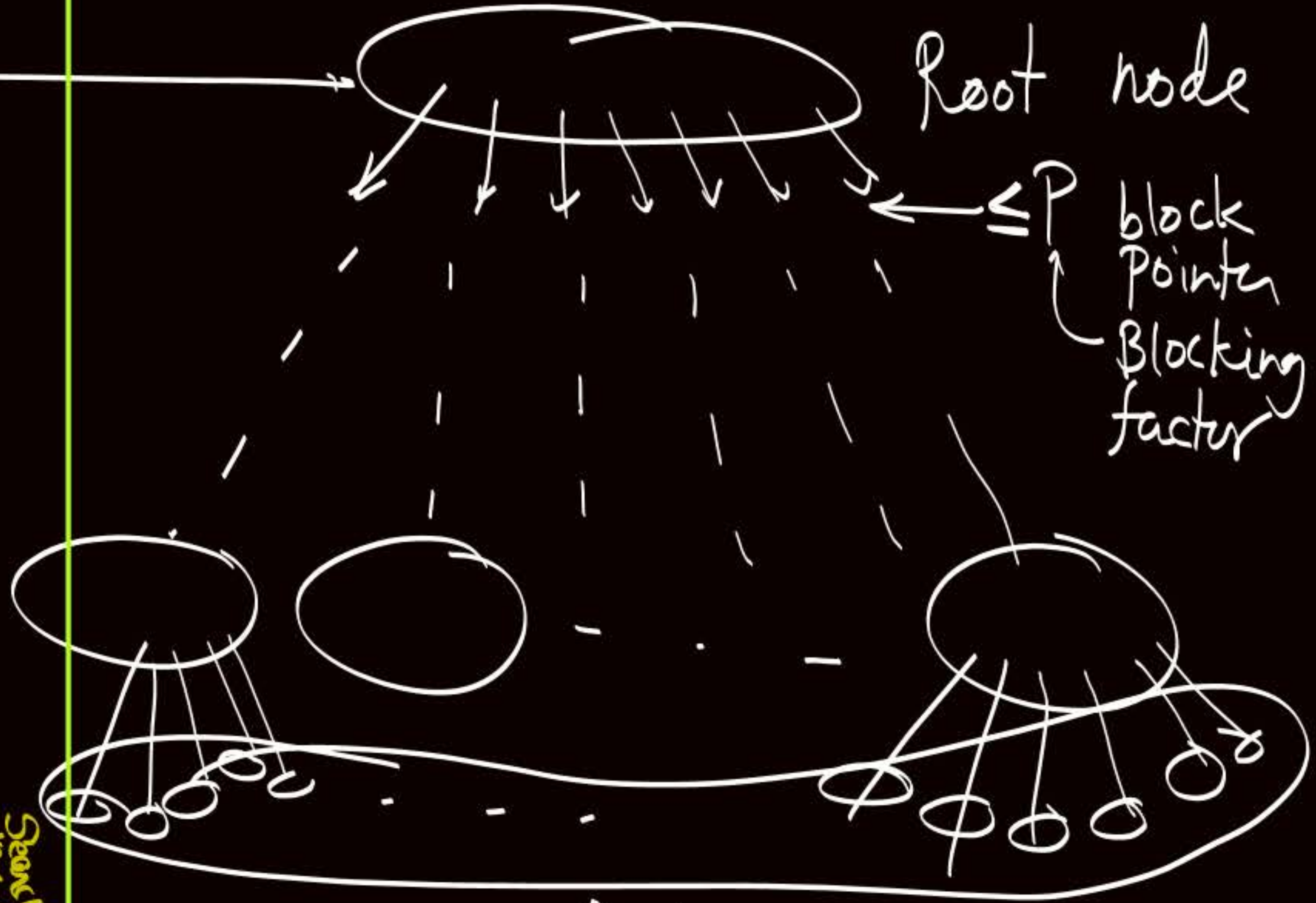


Database file
Search key attribute
other attributes
B1
B2
B3
B4
B5
BN-1
BN
Disk Blocks used to store database file
Search key attribute
Pointer
B1
B2
B3
BM
(M<N) 1st level index
{1st level index may be dense or may be sparse}
Search key attribute
Pointer
B1
B2
Bx
(X<M) 2nd level index
it index the entries of 1st level index
(Sparse index)
Search key attribute
Pointer
B1
Last level of index tree
Last level of index is when all entries can be accomodated into a single Block


From 2nd level onwards all other levels of index are Sparse index
Search key attribute
Pointer
B1
last level of index tree
Last level of index is when all entries can be accomodated into a single Block
Search key attribute
Pointer
B1
B2
Bx
(X<M) 2nd level index
it index the entries of 1st level index
(Sparse index)
Search key attribute
Pointer
B1
B2
B3
BM
(M<N) 1st level index
{1st level index may be dense or may be sparse}
Database file
Search key attribute
other attributes
B1
B2
B3
B4
B5
BN-1
BN
Disk Blocks used to store database file

Multi-level index tree



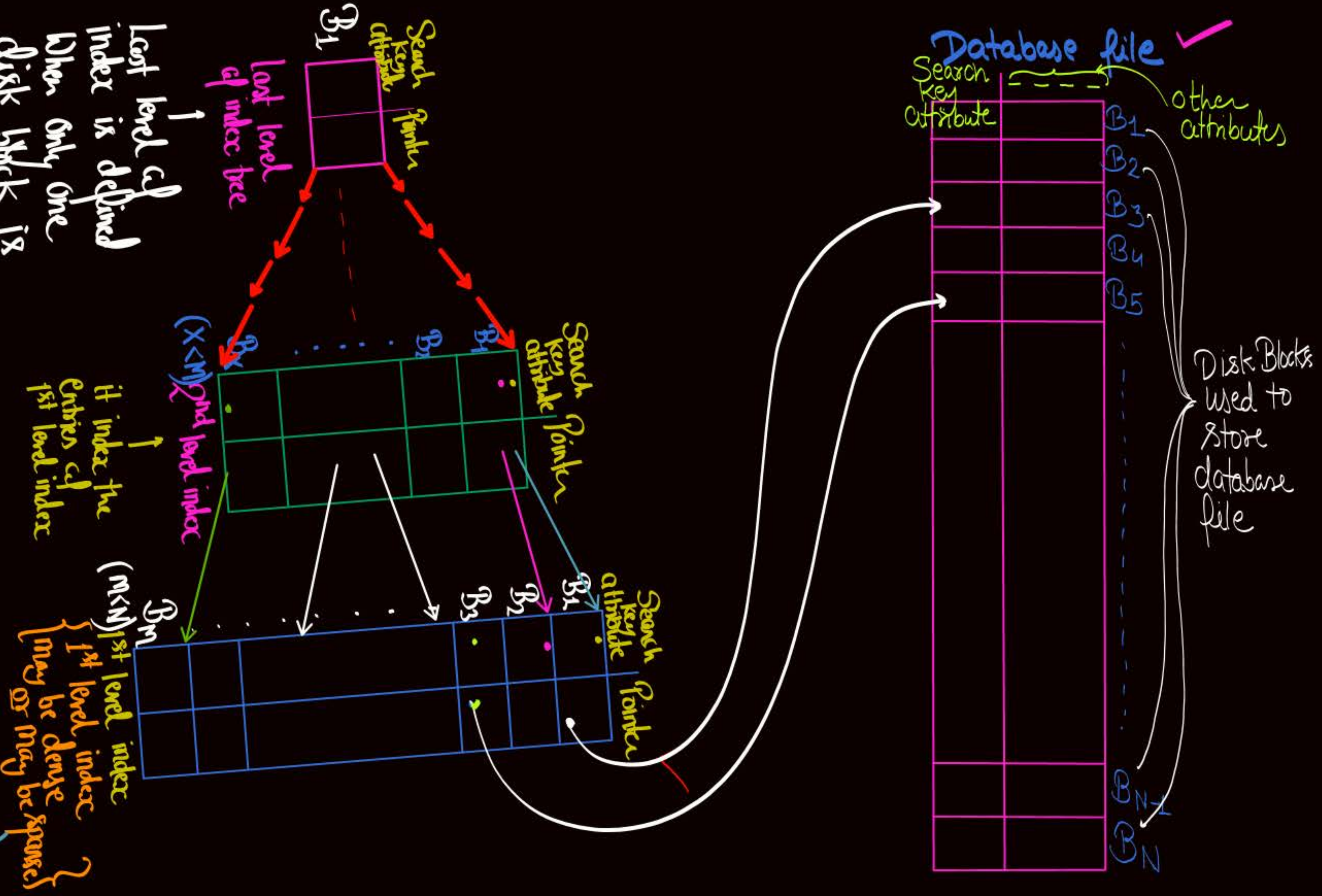


(leaf level of Multi-level index tree)

No. of levels in Multi-level index =

Last level of index is defined when only one disk block is required

(leaf level of Multi-level index tree)



Note :→

IO Cost using multi-level index

IO Cost using multi-level Index = (Number of levels in multi-level index tree) + 1

Number of levels in multi-level index tree $\approx \lceil \log_{Bf} M \rceil$

Bf = Blocking Factor w.r.t. index block (2nd level onwards)

M = No. of disk blocks w.r.t. 1st level index

(Number of levels in multi-level index tree): To locate an entry in the 1st level (leaf level) index. Starting from root level (from each level One block will be transferred)

1: To access the record from the database file based on the address obtained from the Entry of first level index

Note :→

$\boxed{\text{IO Cost using multi-level index}}$

$$\boxed{\text{IO Cost using multi-level Index} = \left(\text{Number of levels in multi-level index tree} + 1\right)}$$

it will be same for all access

i.e, Same IO Cost for all three cases,

Best Case, Worst Case, Average Case

Topic : Multi-level Index

In multi-level index,

1. 1^{st} level of index may be dense or may be sparse
2. Except first level, all other levels of index are always sparse index

- Multi-level index is suitable only for static database i.e. the database in which insertion & deletion are very rare

- But if database is dynamic {i.e. insert and delete opⁿ are frequent} then after every insertion and/or deletion we may have to restructure all the levels of multi-level index. and it will be a time consuming operation

* ∴ for dynamic database we use

→ $\underbrace{\text{B tree or B+ tree}}_{\text{these are self Balancing multi-way search tree}}$

* $\underbrace{\text{Self-balancing multi-way search tree}}$

B tree
B+ tree

These are balanced ← i.e all leaf nodes are at same level

multi-way search tree

↓

m-ary search tree

└── it may be any value ≥ 2

Structure of B tree

Topic : B Tree

* Order of a node of B tree

In general, Order of a node of B tree is defined as the maximum number of Child pointer a node can have

Consider a B tree of order = P,
└ i.e. maximum number of Child pointer a node can have = P
∴ Maximum number of Keys a node of B tree of order 'P' can have = (P-1)

☆ Structure of an internal node of B tree

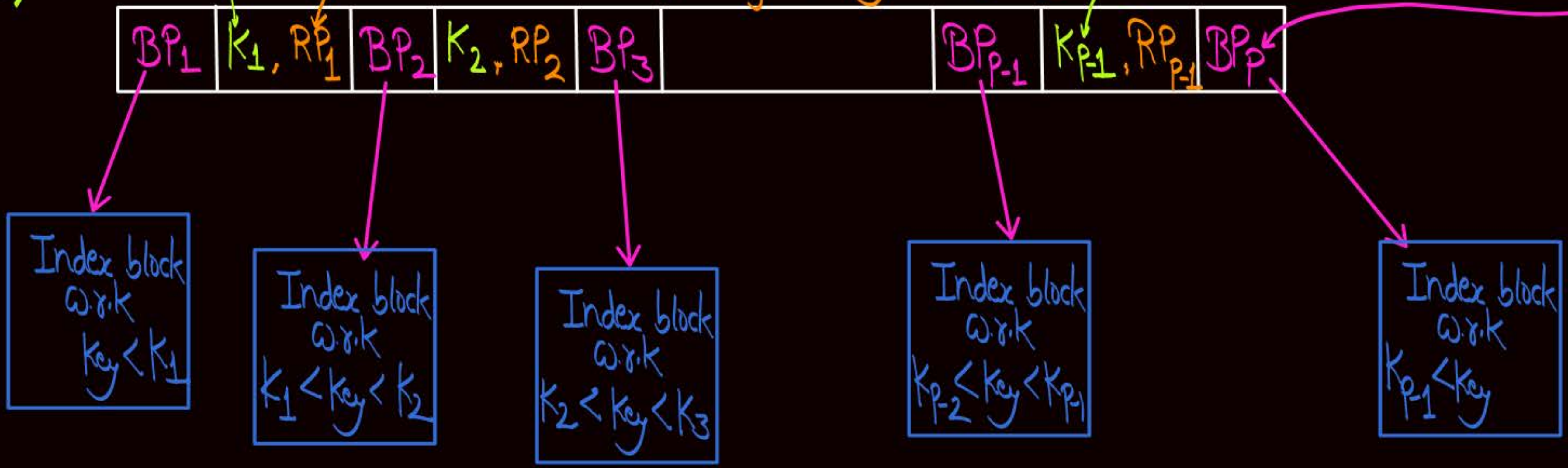
- All keys within the node will always be stored in ascending order
  └ i.e. $K_1 < K_2 < K_3 - - - < K_{P-1}$

✓ One node of B tree →

it is a disk block

Key value

Record Pointer Pointing to the record Corresponding to Key value = K_1



Consider a B tree of order = P,
↳ i.e. maximum number of Child pointer a node Can have = P
∴ Maximum number of Keys a node of B tree of order 'P' Can have = (P-1)

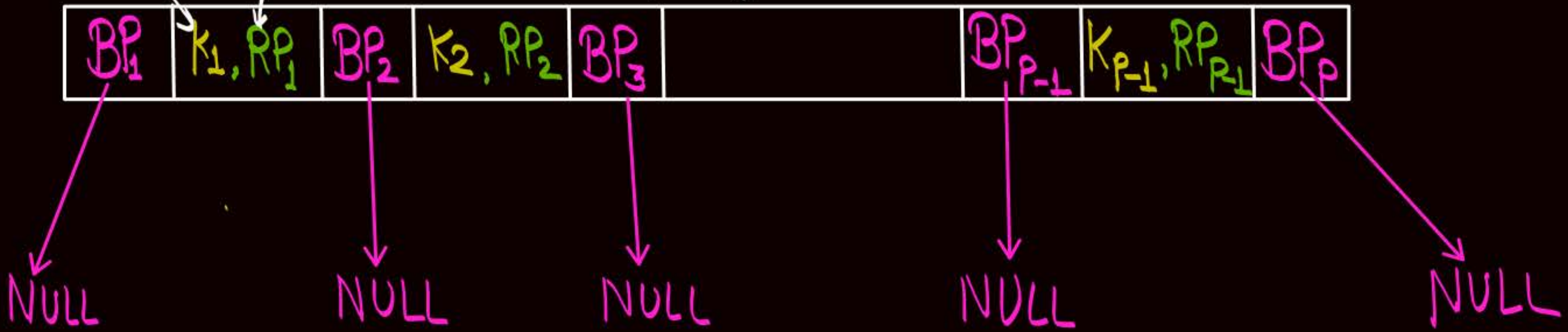
Structure of leaf node of B tree

All keys within the node will always be stored in ascending order
↳ i.e. $K_1 < K_2 < K_3 - - - < K_{P-1}$

Key value = K_1

Record Pointer, pointing to the record Corresponding to Key value = K_1

A node of B tree
↑
it is a block of disk



- BP (Block Pointer) : It is a pointer that points to the index block at lower level of B tree

$\left.\begin{array}{l}\text{Child Pointer}\\ \text{node Pointer}\\ \text{Tree Pointer}\end{array}\right\}$

- RP (Record Pointer) : It Points to the record in the database file corresponding the associated Key value

* **Note :-** In B tree, record pointers are present in internal node as well as in leaf node.
∴ Structure of B tree is different from the structure of multi-level index

* In multi-level index record pointers are present only at leaf level {i.e, 1st level index}

Note:- Each node of B tree is a disk block

∴ whatever information we want to store in one node of B tree must fit in a single disk block.

Consider a B tree of order = P ✓,

↳ i.e. maximum number of Child Pointer a node Can have = P

∴ Maximum number of Keys a node of B tree of order 'P' Can have = (P-1)

Key value = K_1

Record Pointer, pointing to the record Corresponding to Key value = K_1

One node of B tree

↑

One disk block

BP_1	K_1, RP_1	BP_2	K_2, RP_2	BP_3		BP_{P-1}	K_{P-1}, RP_{P-1}	BP_P

$$\left(\left(P * \text{Block Pointer size} \right) + (P-1) * \left(\text{Key field size} + \text{Record Pointer size} \right) \right) \leq \text{Disk Block size}$$

Using this Equation we can determine the maximum order Possible for a node of B tree

Topic : B Tree

Consider a relations R with a key filed K. A B-tree of order P is used to access structures on K. Where 'P' denotes the maximum Number of tree pointers in a B-tree index node. Assume that key (k) is 30 bytes long disk block size 1024 byte. Each data pointer size P_D is 12 bytes long & each block pointer P_B is 10 byte long. In order for each B-Tree Node to fit in a single disk block, the maximum value of P is-

Record

$$\left[\left(P * \text{Block Pointer Size}\right) + (P-1)\left(\text{Key field size} + \text{Record Pointer Size}\right)\right] \leq \text{Disk block Size}$$

$$(P * 10) + (P-1)(30+12) \leq 1024$$

$$52P \leq 1024 + 42$$

$$P \leq \frac{1066}{52} \Rightarrow \boxed{P \leq 20.5} \Rightarrow P_{max} = 20$$

A) 19

B) 20

C) 21

D) 22

Order = Max. no. of child pointer a node of B tree can have

Consider a B tree of order = p

- Maximum number of child pointer a node can have is = p
- Maximum number of keys a node can have is = (p-1)
- Minimum number of child pointer a non-root node must have is = $\lceil p/2 \rceil$
- Minimum number of keys a non-root node must have is = $\lceil p/2 \rceil$-1
- Minimum number of child pointer a root node must have is = 2
- Minimum number of keys root node must have is = 1

Each non-root node of B tree must be at least half full.

Structure of B+ tree

Topic : B+ tree visualization

- B+ tree is some-what similar to multi-level index

In our example we are maintaining Equality Cond^n in right sub-tree

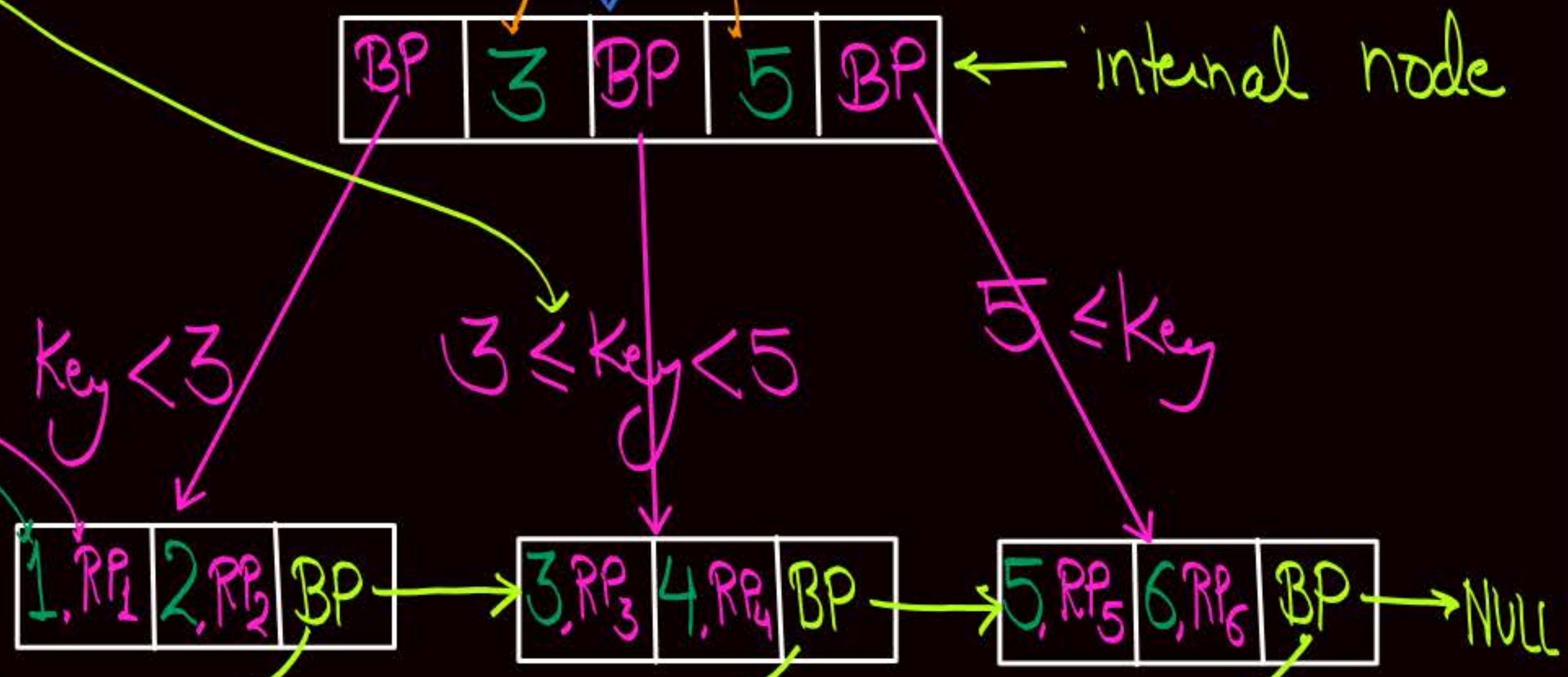
Keys in the internal nodes are dummy keys, they are Used only for search operation

No record Pointers are maintained in internal nodes of B+ tree

Record pointer w.r.t Search key value

Search key value

Nodes at leaf level



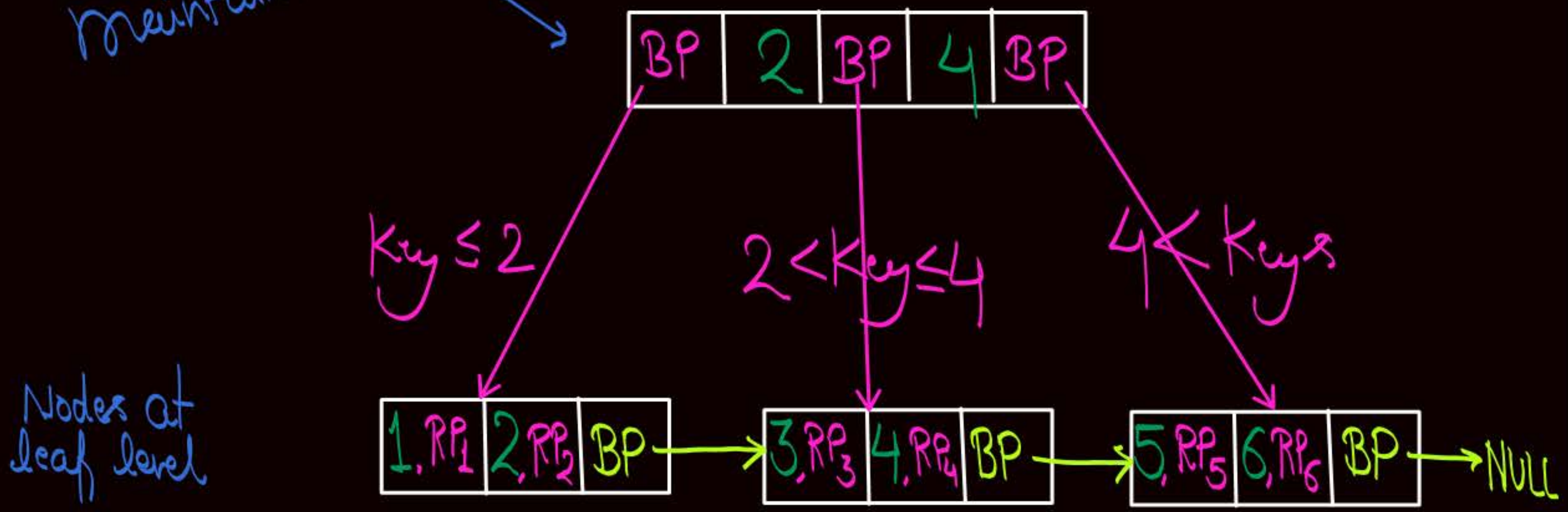
Block pointers Pointing to the next node at leaf level

All the keys of database file are present at leaf level along with their record pointers.

Database file

Search key	- - - - other attributes
2	
1	
3	
5	
6	
4	

Key values in internal node when Equality Condⁿ is maintained in left sub-tree



Database file

Search key	------ other attributes
2	
1	
3	
5	
6	
4	

Topic : B+ Tree

- B+ tree is some what similar to normal multi-level index.
- In B+ Tree all keys along with their record pointers are present at leaf nodes of B+ Tree.
- Keys within a node of B+ tree are always stored in ascending order.
- In B+ Tree keys in the internal nodes are used only for search operation (they are dummy keys), they are not used to point any record of the database as no record pointer is available in internal nodes of B+ Tree.

Structure of an internal node of B+ tree

* All keys within the node are in increasing order

Key value

BP_1		K_1	BP_2	K_2	BP_3	K_3	BP_4	. . . .	BP_{q-1}	K_{q-1}	BP_q

$Key < K_1$ $\quad$ $K_1 \le Key < K_2$ $\quad$ $K_2 \le Key < K_3$ $\quad$ $K_{q-2} \le Key < K_{q-1}$ $\quad$ $K_{q-1} \le Key$

{ let, Equality Cond^n is maintained in right Sub-tree }

Topic : Order of Internal node of B+ Tree

Order of an internal node of B+ tree is defined as the maximum number of child pointer (block pointer/ node pointer) a B+ tree node can have. (same as B Tree)

Let order of internal node of B+ tree is denoted by 'q' { i.e. let internal node can have at most 'q' child pointer, }

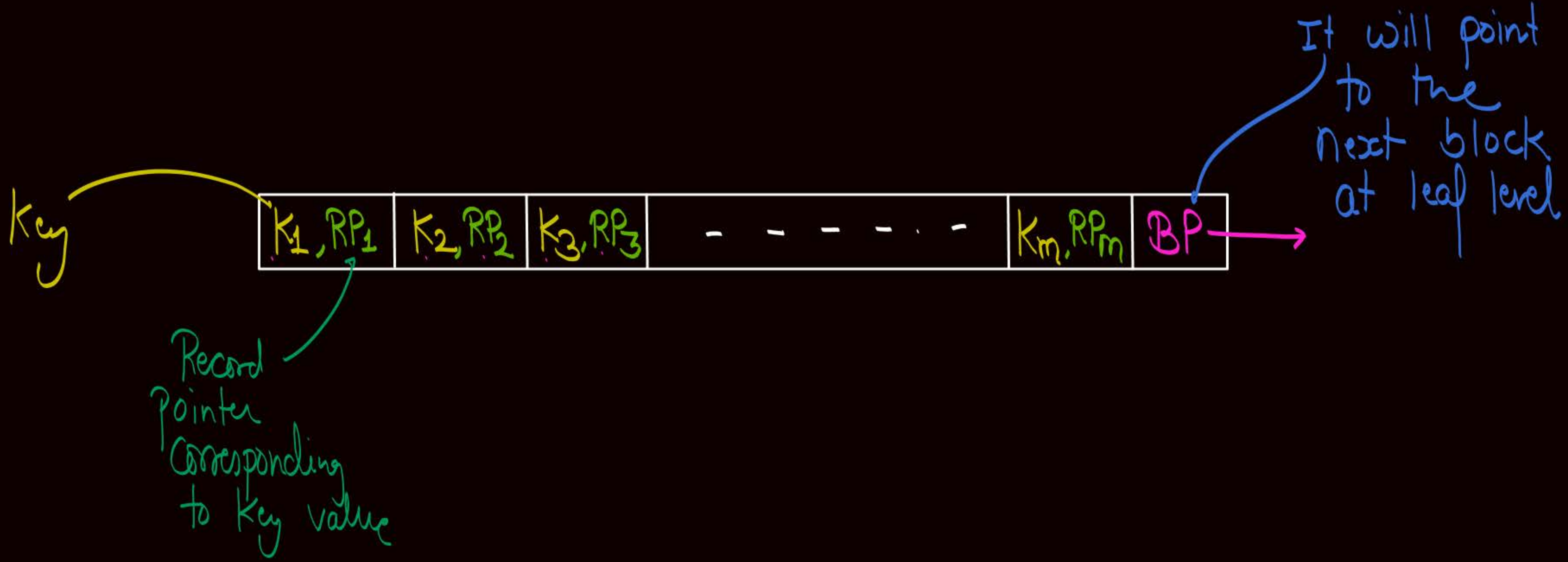
- Whatever information we want to store in an internal node of B+ tree must fit in a single disk block.
- If order of internal node of B+ tree is 'q' then

$$\boxed{\left[\left(q * \underset{\text{size}}{\text{Block Pointer}}\right) + (q-1) * \underset{\text{size}}{\text{Key field}}\right] \leq \underset{\text{size}}{\text{Disk Block}}}$$

Topic : Order of Internal node of B+ Tree

- Let ‘q’ is the order of an internal node of B+ Tree
  - Maximum number of child pointer an internal node can have is = q
  - Maximum number of keys an internal node can have is = q-1
  - Minimum number of child pointer a non-root internal node must have is = $\lceil q/2 \rceil$
  - Minimum number of keys a non-root internal node must have is = $\lceil q/2 \rceil$-1
  - Minimum number of child pointer root node must have is = 2
  - Minimum number of keys root node must have is = 1

Structure of leaf node of B+ tree



Topic : Order of leaf node of B+ Tree

Order of a leaf node of B+ tree is defined as the maximum number of keys (or record pointers) a B+ tree leaf node can have. [redacted]

Let order of a leaf node of B+ tree is denoted by 'm'

Whatever information we want to store in a leaf node of B+ tree must fit into a single disk block

If order of leaf node of B+ tree is given as 'm', then

$$\boxed{\left[m * (\text{Key field size} + \text{Record pointer size}) + (1 * \text{Block pointer size})\right] \leq \text{Disk Block size}}$$

Topic : Order of leaf node of B+ Tree

Let 'm' is the order of a leaf node of B+ Tree

- Maximum number of keys a leaf node can have is = m
- Minimum number of keys a non-root leaf node must have is = $\lceil m/2 \rceil$

it should be at least half full

- If leaf node is the root node as well, then is must contain minimum '1' key.
- There will be exactly one block pointer in each leaf node, it will point to the next leaf node of B+ tree. (because of this pointer range query is more efficient in B+ tree)

eg: Identify records of Employees from Eid = 100 to Eid = 200

Topic : NOTE

- If only one order is given in the question, "w.r.t. B+ tree" then we will use the same order for both leaf node as well as for internal node, but it will be handled differently

Imp:-

Consider order of node of B+ tree is = n

{only one order is given}

Korth (w.r.t. Internal node)

* Maximum number of Child Pointer an internal node of B+ tree can have = n
* Maximum number of Keys an internal node of B+ tree can have = $(n-1)$
* Minimum number of Child Pointer a non-root internal node of B+ tree must have = $\lceil \frac{n}{2} \rceil$
* Minimum number of Keys a non-root internal node of B+ tree must have = $\left(\lceil \frac{n}{2} \rceil - 1\right)$

for root, Min. no. of Keys = 1
Min. no. of Child ptr = 2

Navathe (w.r.t. Internal node)

* Maximum number of Child Pointer an internal node of B+ tree can have = n
* Maximum number of Keys an internal node of B+ tree can have = $(n-1)$
* Minimum number of Child Pointer a non-root internal node of B+ tree must have = $\lceil \frac{n}{2} \rceil$
* Minimum number of Keys a non-root internal node of B+ tree must have = $\left(\lceil \frac{n}{2} \rceil - 1\right)$

for root, Min. no. of Keys = 1
Min. no. of Child ptr = 2

Imp:-

$\boxed{\text{Consider order of node of B+ tree is} = n}$ {Only one order is given}

Korth (w.r.t. leaf node)

* Maximum number of key a leaf node of B+ tree Can have $= (n-1)$

* Minimum number of keys a non-root leaf node must have $= \left\lceil \frac{(n-1)}{2} \right\rceil$

Navathe (w.r.t. leaf node)

* Maximum number of key a leaf node of B+ tree Can have $= (n-1)$

* Minimum number of keys a non-root leaf node must have $= \left\lceil \frac{n}{2} \right\rceil$

Practice questions

#Q.

Assume 'Root' is at "height = 0" or "level = 1"

a) Find the minimum number of nodes required in B tree of order = p and height = h

b) Find the minimum number of child pointers required in B tree of order =p and height = h

c) Find the minimum number of keys required in B tree of order=p and height = h

d) Find the maximum number of nodes possible in B tree of order = p and height = h

e) Find the maximum number of child pointers possible in B tree of order = p and height = h

f) Find the maximum number of keys possible in B tree of order = p and height = h

Analysis w.r.t. B tree of Order = P

Height/ level	Minimum number of nodes at	minimum number of Child pointer at	Minimum number of Keys/RP at	Maximum number of nodes at	Maximum number of Child pointer at	Maximum number of Keys/RP at
(root) $h=0$ / $l=1$	1	2	$(2-1)=1$	1	P	$(P-1)$
$h=1$ / $l=2$	2	$2*\lceil \frac{p}{2} \rceil$	$2*(\lceil \frac{p}{2} \rceil -1)$	P	$P*P=P^2$	$P*(P-1)$
$h=2$ / $l=3$	$2*\lceil \frac{p}{2} \rceil$	$(2*\lceil \frac{p}{2} \rceil)*\lceil \frac{p}{2} \rceil$ $=2*\lceil \frac{p}{2} \rceil^2$	$(2*\lceil \frac{p}{2} \rceil)(\lceil \frac{p}{2} \rceil -1)$	P^2	$P^2*P=P^3$	$P^2*(P-1)$
$h=3$ / $l=4$	$2*\lceil \frac{p}{2} \rceil^2$	$2*\lceil \frac{p}{2} \rceil^2*\lceil \frac{p}{2} \rceil$ $=2*\lceil \frac{p}{2} \rceil^3$	$(2*\lceil \frac{p}{2} \rceil)^2*(\lceil \frac{p}{2} \rceil -1)$	P^3	P^4	$P^3(P-1)$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Height $=h$ / $l=h+1$	$2*\lceil \frac{p}{2} \rceil^{h-1}$	$2*\lceil \frac{p}{2} \rceil^{h}$	$(2*\lceil \frac{p}{2} \rceil)^{h-1}*(\lceil \frac{p}{2} \rceil -1)$	P^h	P^{h+1}	$P^h(P-1)$

Q: Min. no. of nodes required in a B tree of order = p and height = h {where root is at height = 0}

$$= 1 + 2 + 2*\left\lceil\frac{p}{2}\right\rceil + 2*\left\lceil\frac{p}{2}\right\rceil^2 + 2*\left\lceil\frac{p}{2}\right\rceil^3 + \cdots + 2*\left\lceil\frac{p}{2}\right\rceil^{h-1}$$

$$= 1 + \underbrace{2*\left\lceil\frac{p}{2}\right\rceil^0 + 2*\left\lceil\frac{p}{2}\right\rceil + 2*\left\lceil\frac{p}{2}\right\rceil^2 + 2*\left\lceil\frac{p}{2}\right\rceil^3 + \cdots + 2*\left\lceil\frac{p}{2}\right\rceil^{h-1}}_{\text{GP with 1st term = 2, Common ratio} = \left\lceil\frac{p}{2}\right\rceil \text{ \& No. of terms} = h}$$

$$= \boxed{1 + \frac{2*\left(\left\lceil\frac{p}{2}\right\rceil^h - 1\right)}{\left(\left\lceil\frac{p}{2}\right\rceil - 1\right)}}$$

Summation of GP $= \dfrac{a\,(r^n - 1)}{(r - 1)}$

H.W.

#Q. Consider a file of 10,00 records, and disk blocks of 1000 bytes. There are two alternate options to create a single level index file:

Option 1: Index file is created on an unordered key field , where key field is 12 bytes long.

Option 2: Index file is created on an ordered nonkey field, where non-key field is 20 bytes long.

Block pointer size is 10 bytes long and unspanned organization is used. & Record Pointer size is 15 bytes long.

Let 'X' is the blocking factor of index block using option 1 and 'Y' is the blocking factor of index block using option 2 then |X – Y| is ______.

H.W.

#Q. [redacted] Consider an unordered file of 10,0000 records with a record size of 100 bytes stored on blocks of 1KB with an unspanned record organization. We will assume that no system related information is stored within a block. How many blocks would be needed to store this file? _______

HW:

#Q.[redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is ordered by the key field Eid and we want to construct a primary index on Eid.

"By default it is Sparse"

The record size of the file is______bytes

H.W.

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is ordered by the key field Eid and we want to construct a $\underbrace{\text{primary index on}}_{\text{Sparse}}$ Eid.

The number of disk blocks required to store the file assuming an unspanned organization is used______

H.W:

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is ordered by the key field Eid and we want to construct a $\underbrace{\text{primary index}}_{\text{Sparse}}$ on Eid.

The blocking factor of index block is_____

H.W.

#Q.[redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is Unordered by the key field Eid and we want to construct a secondary index on Eid.

(always dense)

The blocking factor of index block is_____

H.W.

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is ordered by the key field Eid and we want to construct a primary index on Eid.

Total number of entries in first level index is_____

H.W:

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is Unordered by the key field Eid and we want to construct a secondary index on Eid.

(always dense)

Total no. of Entries in 1st level index is____

H.W.

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is ordered by the key field Eid and we want to construct a primary index on Eid.

Total number of disk blocks required to store first level index is_____

H.W.

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is Unordered by the key field Eid and we want to construct a secondary index on Eid.

Total number of disk blocks required to store first level index is_____

HW

#Q [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is ordered by the key field Eid and we want to construct a primary index on Eid.

If we decide to create a multi-level index, then total number of levels needed in multi-level index is______

HW

#Q[redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is Unordered by the key field Eid and we want to construct a Secondary index on Eid.

If we decide to create a multi-level index, then total number of levels needed in multi-level index is______

H.W.

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is ordered by the key field Eid and we want to construct a primary index on Eid.

If we decide to create a multi-level index, then total number disk blocks required to store multi-level index is_____

H.W.

#Q. [redacted] Consider a disk with block size B = 512 bytes. A block pointer is PB = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has 30,000 EMPLOYEE records of fixed length. Each record has the following fields: Ename (30 bytes), Eid (9 bytes), Did(9 bytes), Location (40 bytes), Contact (9 bytes), DOB(8 bytes), Gender (1 byte), Role(4 bytes), and Salary (4 bytes) Other than the specified fields, an additional byte is used as a deletion marker in each record. Suppose the file is Unordered by the key field Eid and we want to construct a secondary index on Eid.

If we decide to create a multi-level index, then total number disk blocks required to store multi-level index is_____

H.W.

#Q. Consider a B tree in which the maximum numbers of keys in a node is '9', The minimum numbers of keys a non-root may have is _______

H.W

#Q. [redacted] Let 'X' denotes the order of an internal node of B+tree, and 'Y' denotes order of a leaf node of B+tree.

If disk block size is 1024 bytes, search key is 15 bytes long, block pointer is 10 bytes long and record pointer is 10 byte long then maximum value of X+Y is_______

HW:

#Q. Let 'X' denotes the order of an internal node of B tree, and 'Y' denotes order of a leaf node of B tree.

If disk block size is 1024 bytes, search key is 15 bytes long, block pointer is 10 bytes long and record pointer is 10 byte long then maximum value of X+Y is_______

H.W.

#Q. Consider the B tree of order 10. Assume that the number of levels in the tree is four including root. The maximum number of record pointers that can be stored in B tree is ____________

H.W.

#Q. [redacted] Consider the B tree of order 10. Assume that the number of levels in the tree is four including root. The minimum number of record pointers that can be stored in B tree is ___________

H.W.

#Q. Consider the B+tree of order 10. Assume that the number of levels in the tree is four including root. The maximum number of record pointers that can be stored in B+tree is

H.W.

#Q. Consider the B+tree of order 10. Assume that the number of levels in the tree is four including root. The minimum number of record pointers that can be stored in B+tree is ____________

2 mins Summary

Topic Structure of B+ tree

Topic [redacted]

Topic [redacted]

Slide

THANK - YOU

Slide