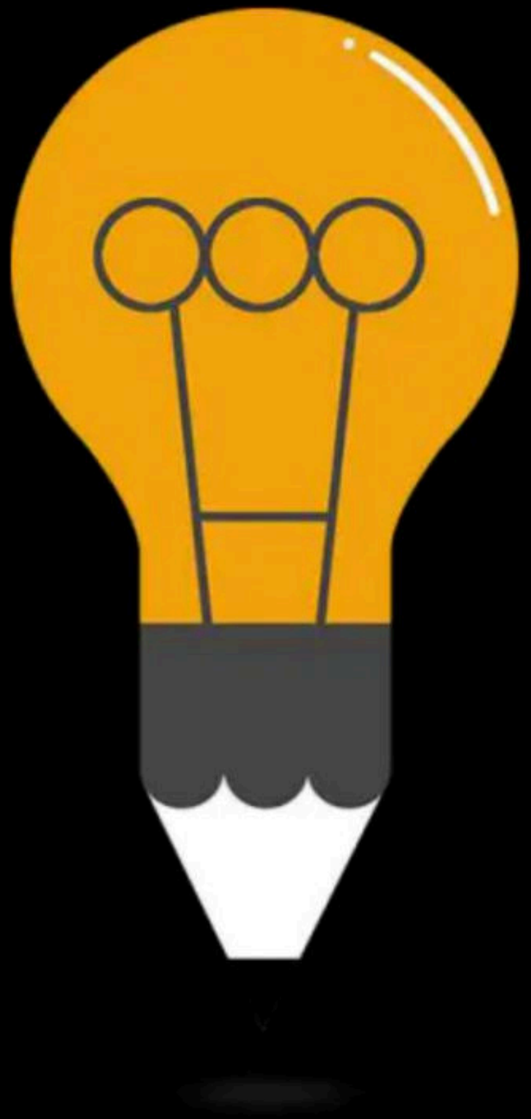
unacademy

# File Organization and Indexing: Part IV

Complete Course on Database Management System

**Vishvadeep Gothi** • Lesson 33 • July 29, 2023

unacademy



# DBMS

# Indexing: B-Tree 2

By: Vishvadeep Gothi

unacademy

# B-Tree

- Tree based indexing
- Dynamic Indexing technique
- Based on insertion and deletion, the tree automatically adjusted
- Self balancing search tree

unacademy

# Binary Search Tree

unacademy

# B-Tree

An order-p B-tree:

1. Every node other than root should have atleast $\left\lceil \frac{p}{2} - 1 \right\rceil$ ~~nodes~~ keys
2. In every node there are atmost (p-1) keys and n tree pointers
3. Root can have minimum 1 node
4. All leaves appear on the same level

unacademy

# B-Tree Node Structure

- Key
- Record Pointer
- Tree Pointer

unacademy

# Insertion in B-Tree

- B-tree of order-3
- Insert keys 1, 2, 3, 4, 5, 6, 7

unacademy

# Insertion in B-Tree

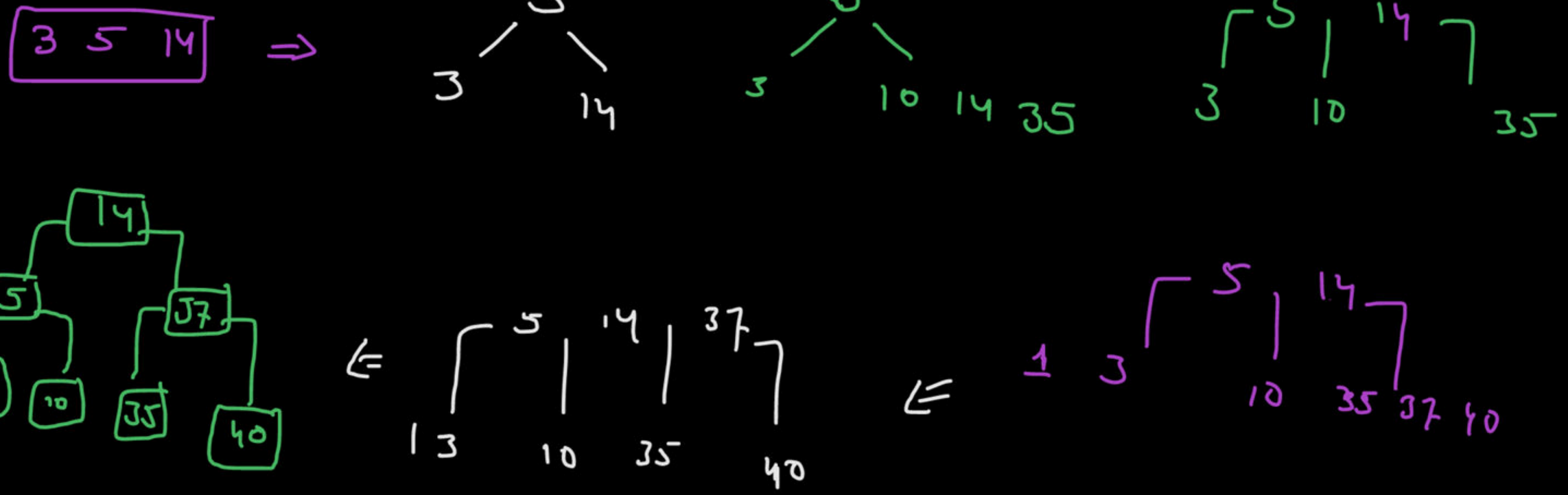
- B-tree of order-5
- Insert keys 7, 4, 14, 25, 3, 10, 12, 15, 17, 9, 29, 1, 38, 3, 11

unacademy

# Insertion in B-Tree

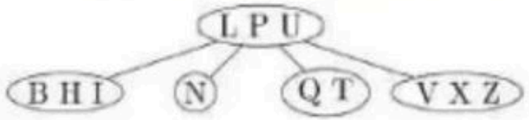
| keys | min | max |
|---|---|---|
| | 1 | 2 |

- B-tree of order-3
- Insert keys 14, 3, 5, 10, 35, 40, 1, 37

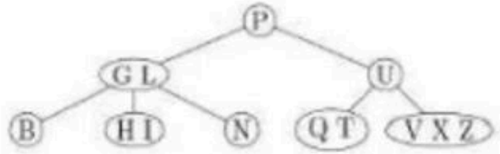


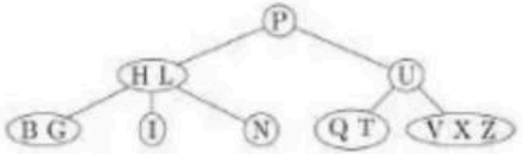
# Question GATE-2003

Consider the following $2-3-4$ tree (i.e., B-tree with a minimum degree of two) in which each data item is a letter. The usual alphabetical ordering of letters is used in constructing the tree.




What is the result of inserting $G$ in the above tree?

A.



B.


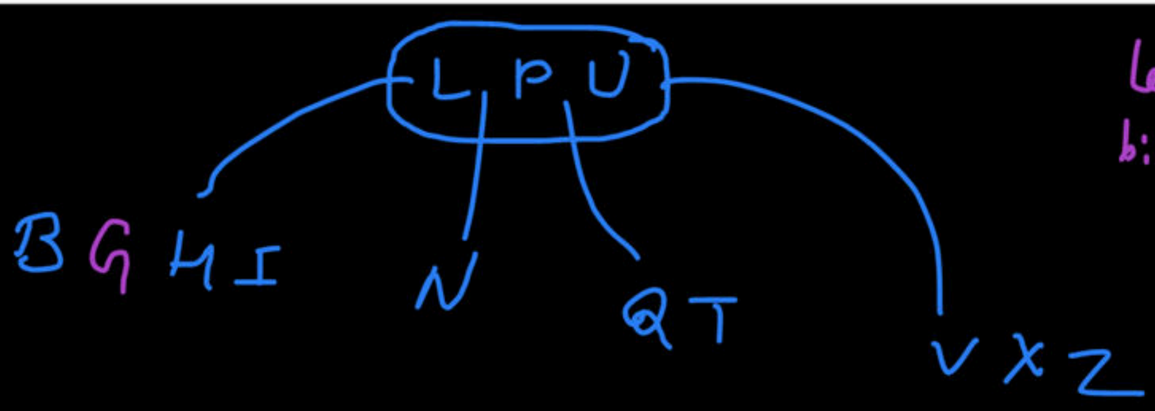

C.



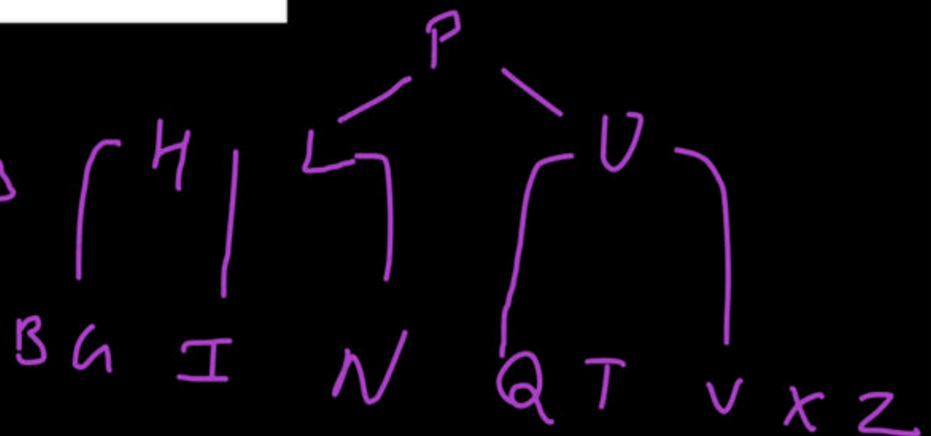
D. None of the above

order 4 B-tree

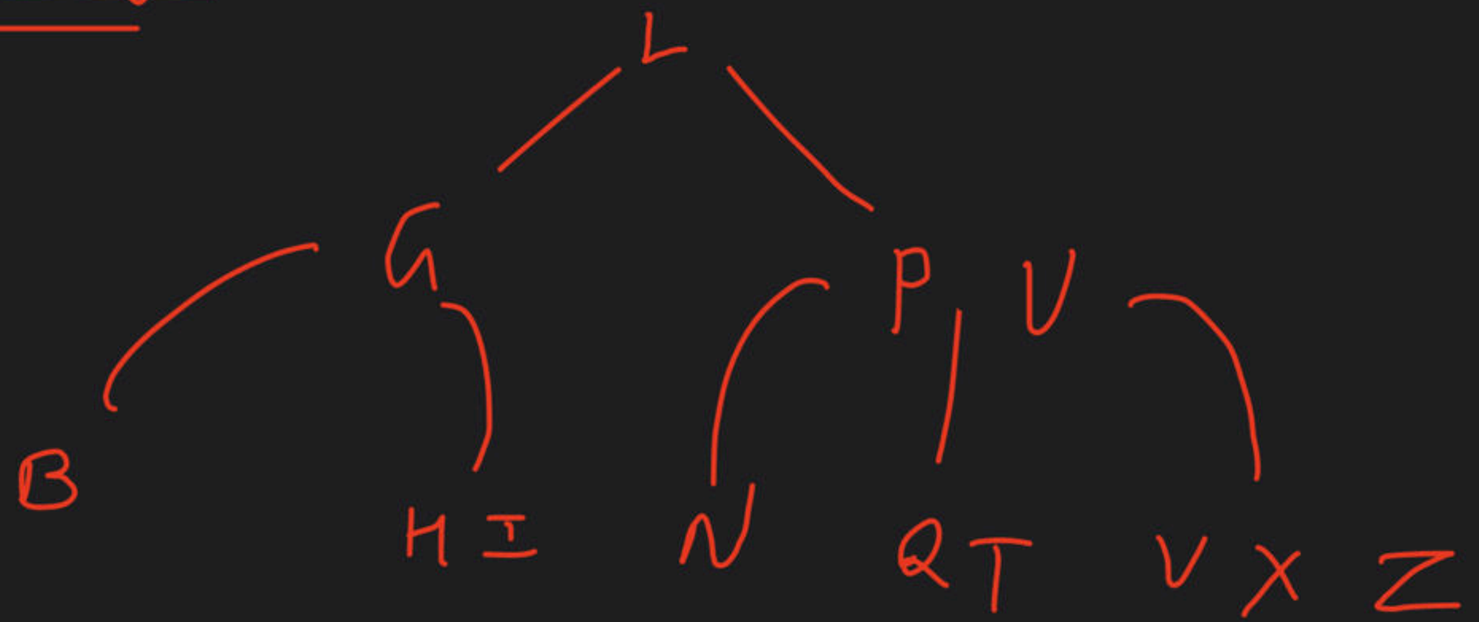
| | Pointers |
|---|---|
| max ⇒ 3 | 4 |
| min ⇒ 1 | 2 |




left biasing ⇒

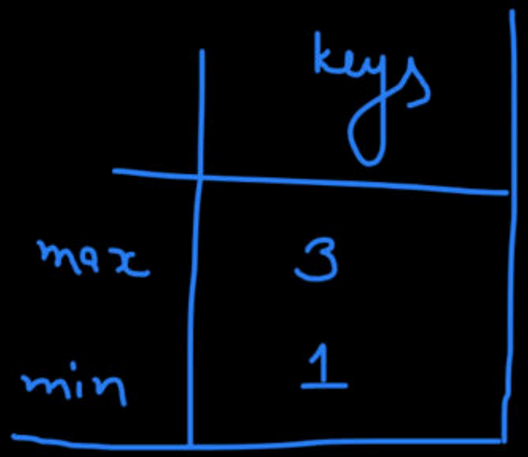


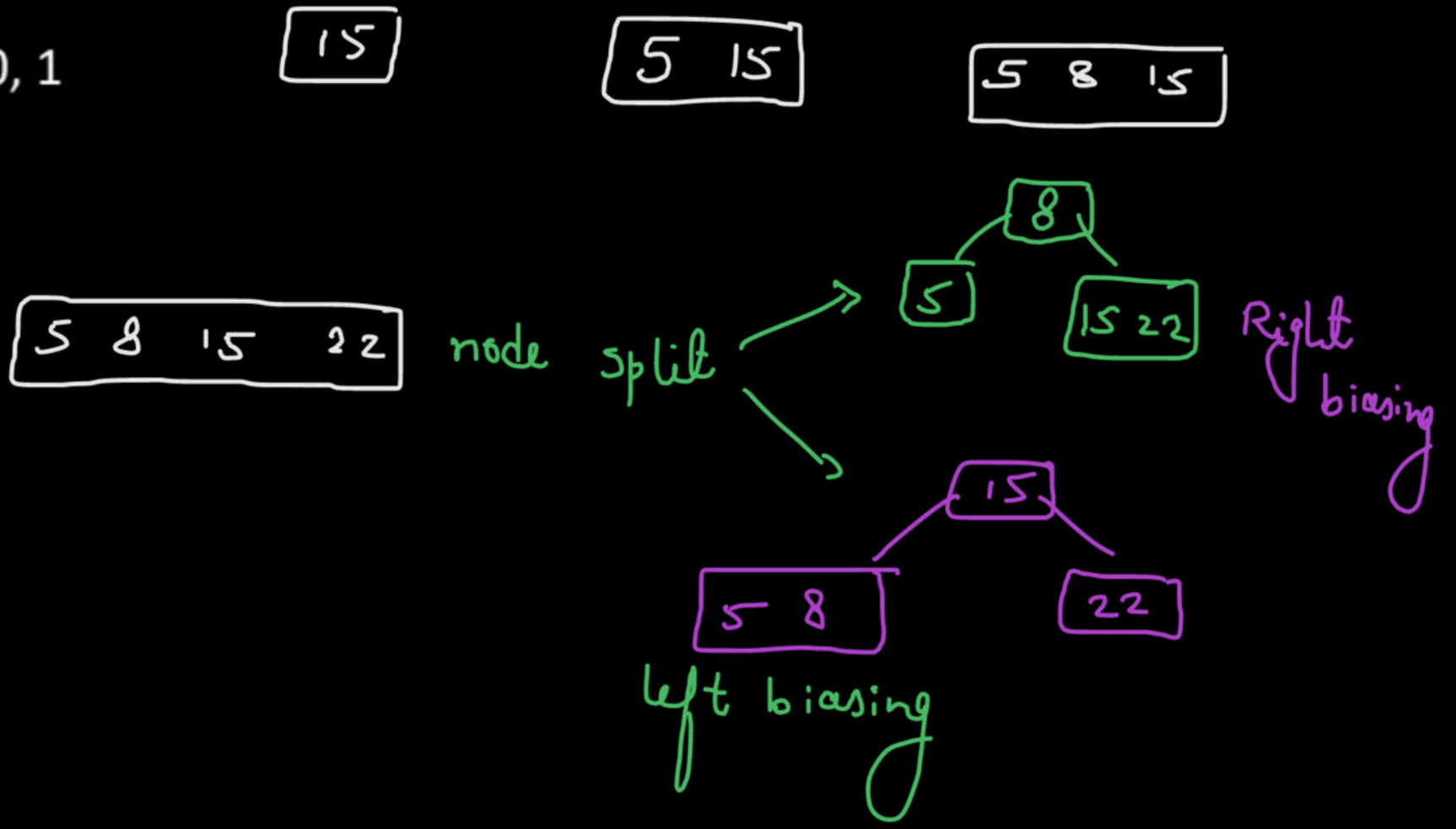
Right hand binary :-



# Insertion in B-Tree

- B-tree of order-4
- Insert keys 15, 5, 8, 22, 10, 1



unacademy

# Insertion in B-Tree

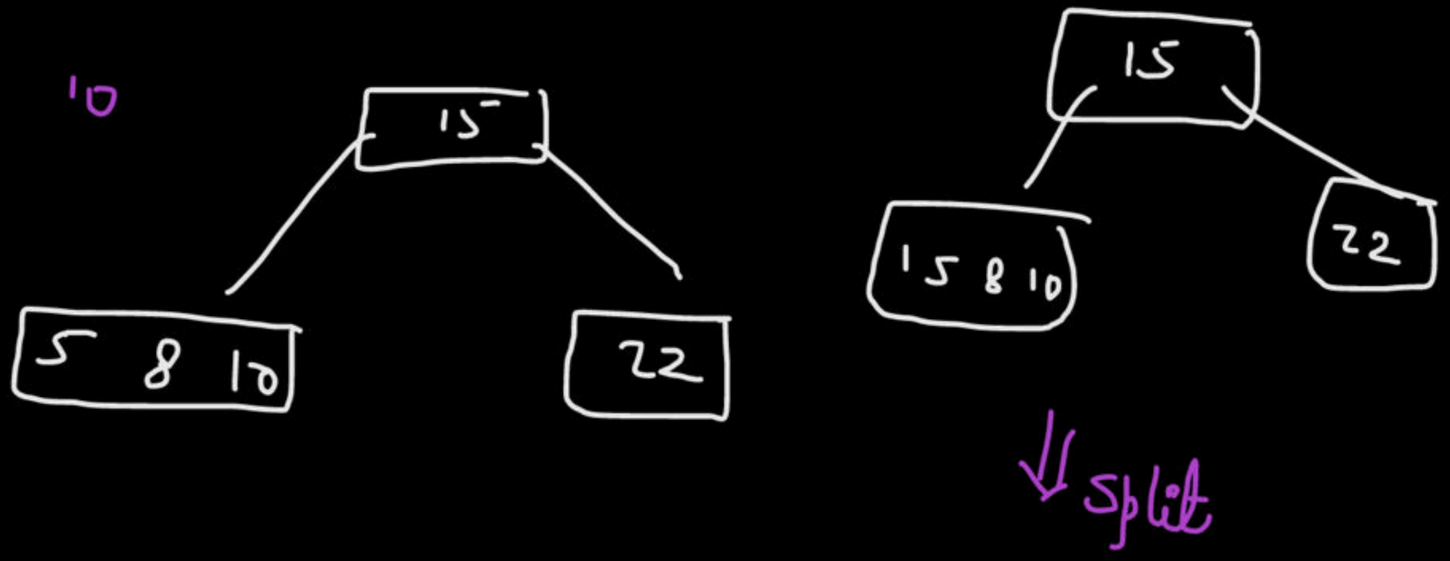
~~More~~ split in left-biasing or right-biasing?

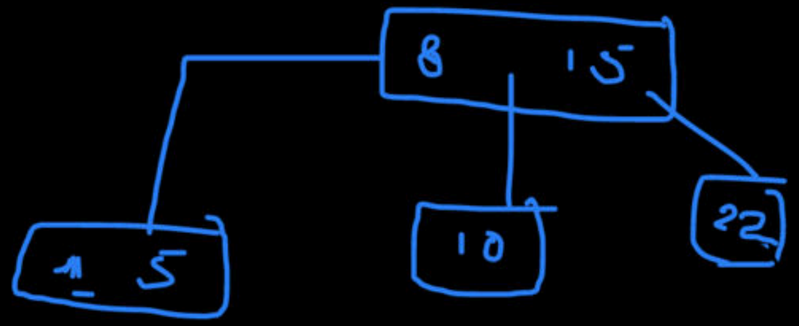
node

Continue with left biasing

Insert 10

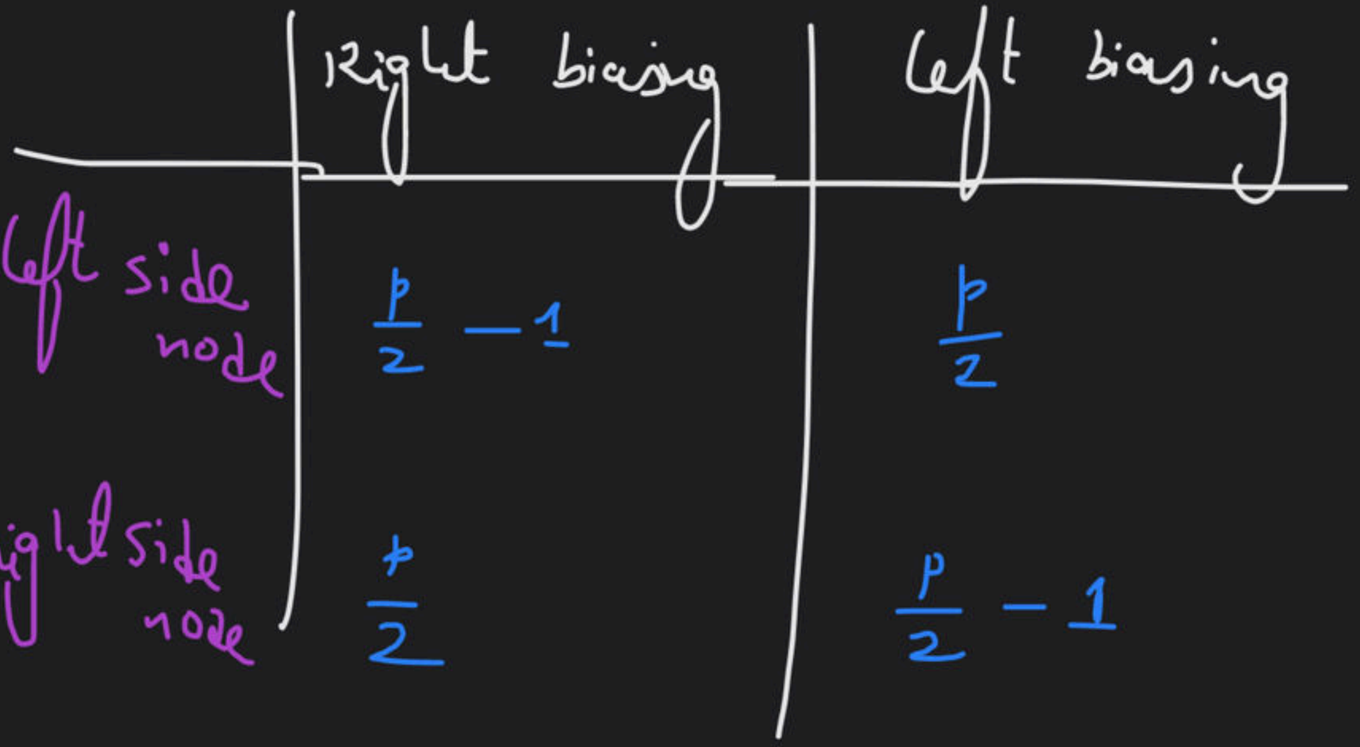
⇒



Assume a B-Tree of order-p  p ⇒ even & p > 2

when during an insertion a node is splitted then the no. of keys in a splitted nodes should be?



# Question GATE-2008

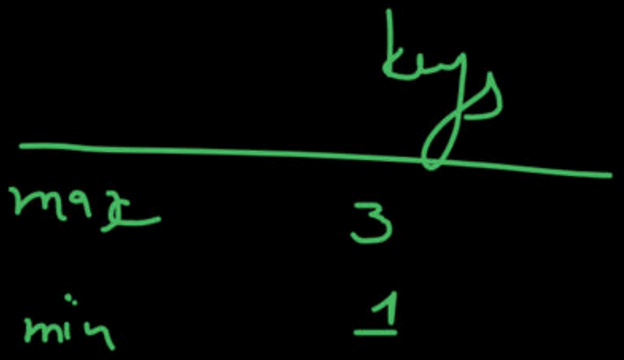
A B-tree of order 4 is built from scratch by 10 successive insertions. What is the maximum number of node splitting operations that may take place?

(A) 3

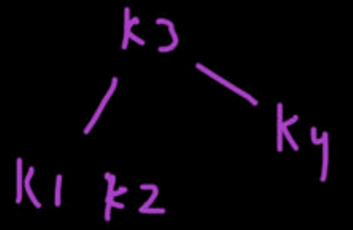
(B) 4

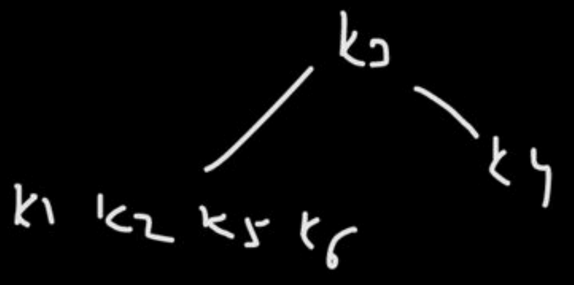
(C) 5

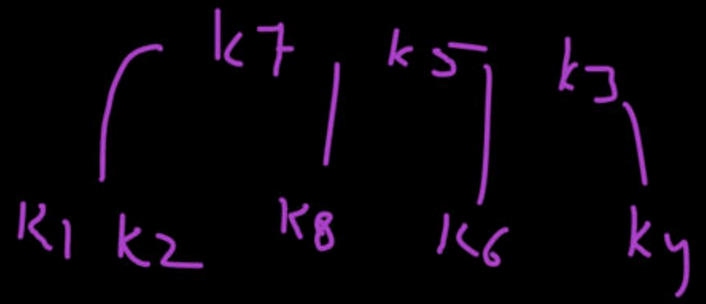
(D) 6

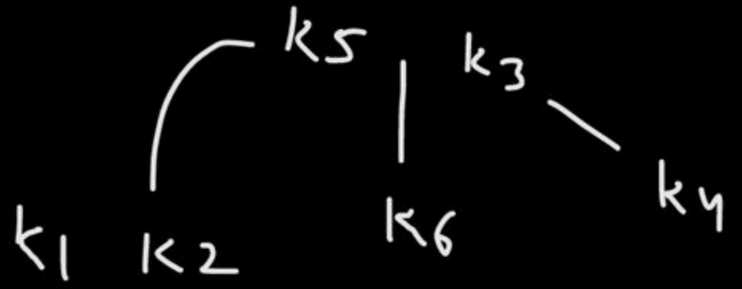






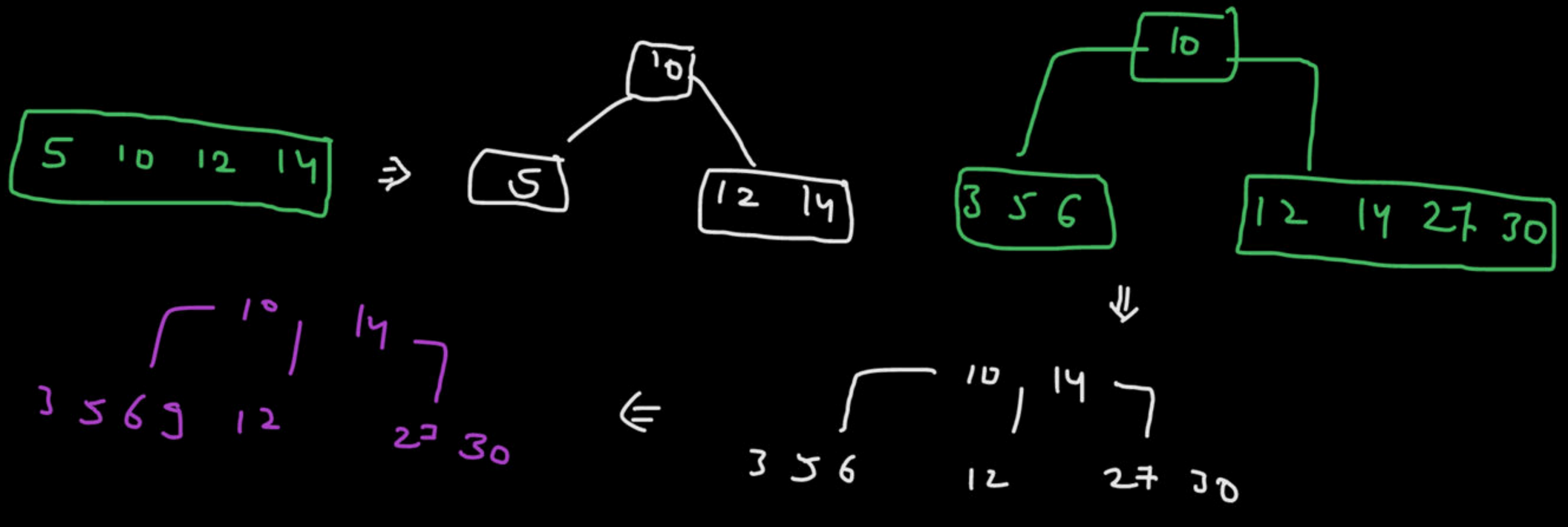
# Question

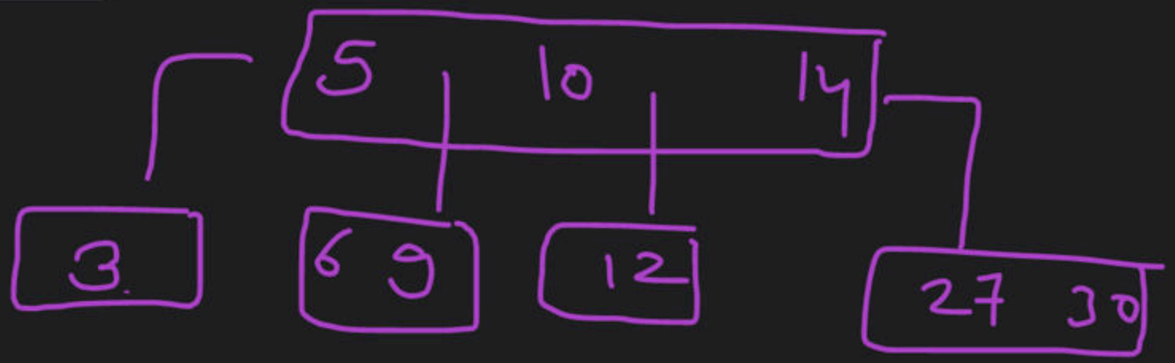
Ans => 1

A B-tree of order 4 is built from scratch by successive insertions of following keys in the given order.

10, 5, 14, 12, 3, 6, 30, 27, 9

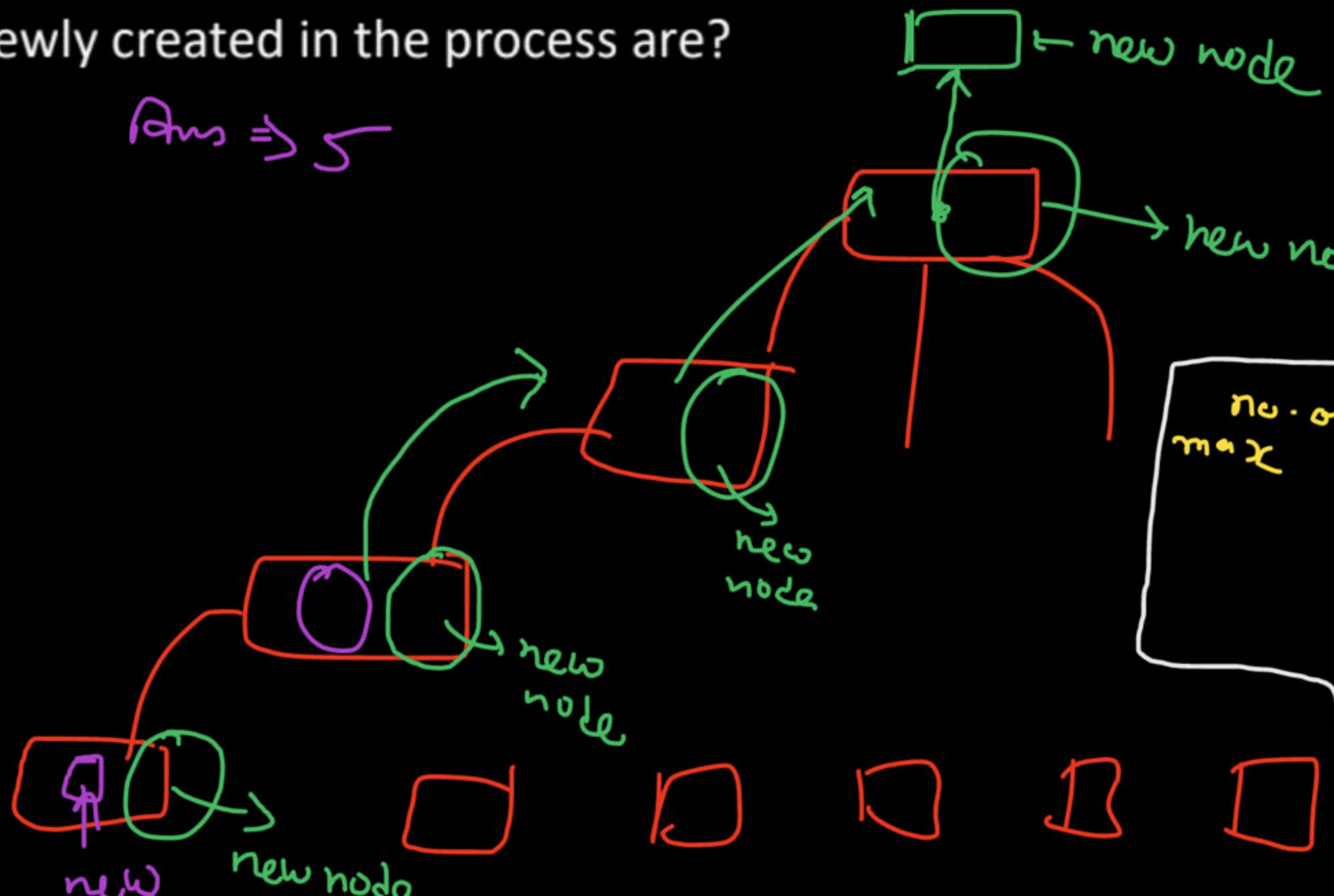
What is the number of root node splitting operations that may take place with right biasing?





# Question GATE-2005

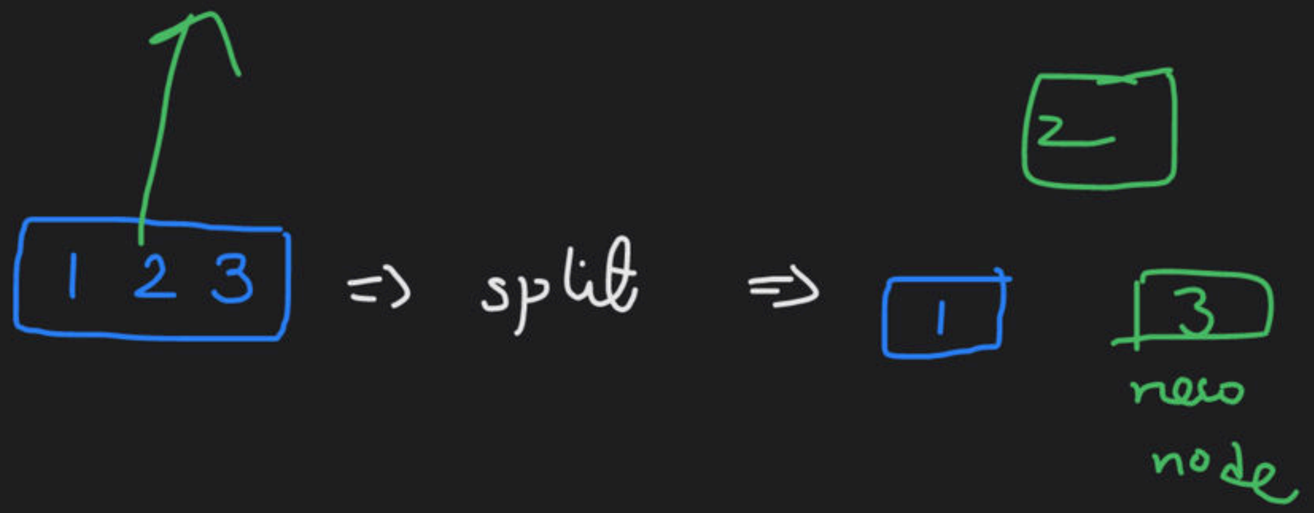
A B-Tree used as an index for a large database table has four levels including the root node. If a new key is inserted in this index, then the maximum number of nodes that could be newly created in the process are?




Ans ⇒ no. of levels + 1

max no. of splits in this questn = no. of levels

order 3 :-



unacademy

# Question GATE-2004

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 a B-tree index node. Assume that K is 10 bytes long; disk block size is 512 bytes; each data pointer PD is 8 bytes long and each block pointer PB is 5 bytes long. In order for each B-tree node to fit in a single disk block, the maximum value of p is?

Ans = 23

$$\left((p-1)\,10 + (p-1)8 + p*5\right) \leq 512$$

$$18p - 18 + 5p \leq 512$$

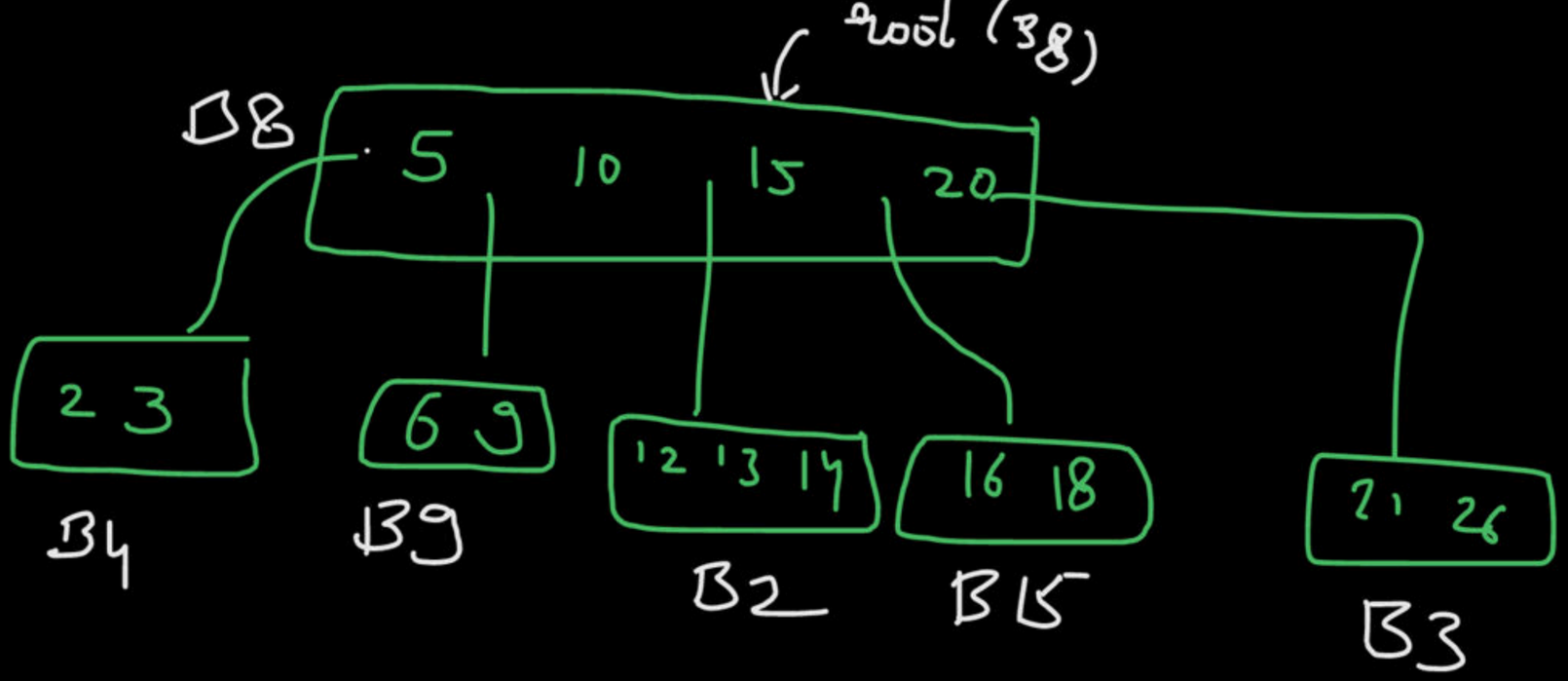
$$p \leq 23.04 \Rightarrow \boxed{p_{max} = 23}$$

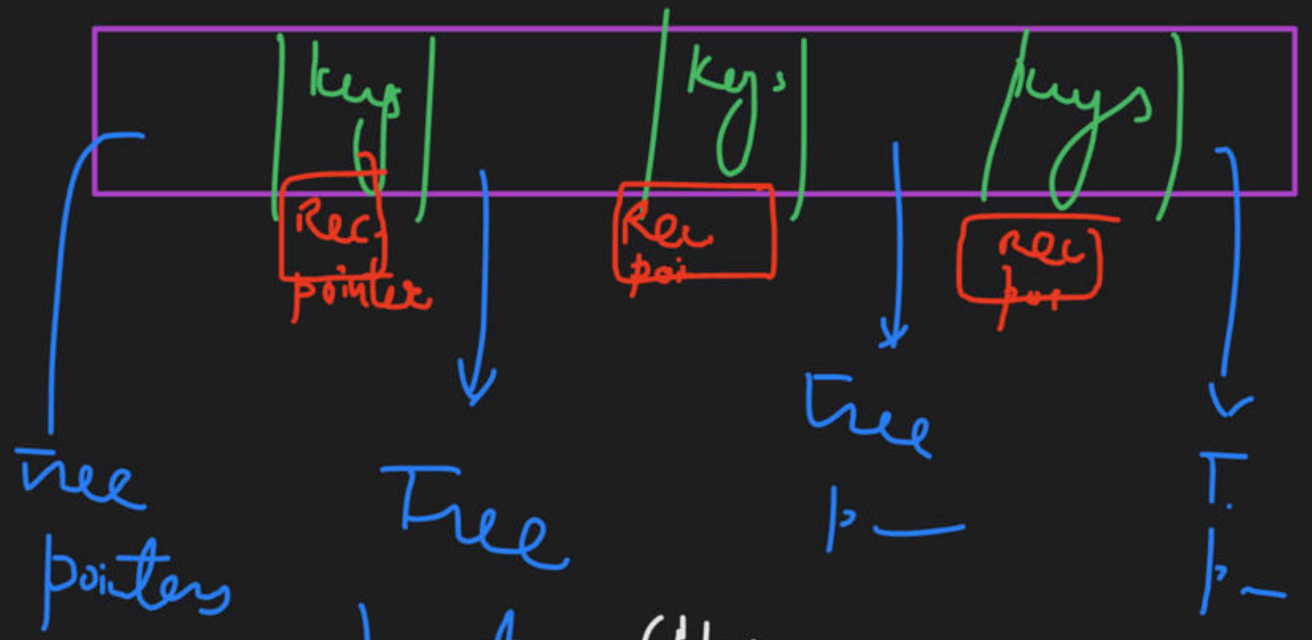
# Practical Implementation of Node on Blocks

What is maximum order in B-tree?

Idea to store B-Tree on disk => keep one node of tree on a block

ex:- order-5 B-Tree






(block no. or block pointer)

max keys $\Rightarrow p - 1$

max tree pointers $\Rightarrow p$

max rec. pointers $\Rightarrow p - 1$

$$\left[ (p-1) \text{ keys size} + (p-1) \underset{\text{size}}{\text{rec. pointers}} + p * \underset{\text{size}}{\text{tree pointers}} \right] \leq \text{block size}$$

# Question

Key size = 16 bytes

Block pointer size = 32 bytes

Record pointer size = 48 bytes

Block size = 8192 bytes

If a B-tree of order-p is implemented, then what is the maximum value of p?

Ans => 86

$$\left( (p-1) * 16 + (p-1)\,48 + p * 32 \right) \leq 8192$$

$$64p - 64 + 32p \leq 8192$$

$$p \leq 86 \Rightarrow \boxed{p_{max} = 86}$$

unacademy

# Height of the B-tree

P-order B-tree

Total nodes = n

$$H_{min} = \lceil \log_p(n+1) - 1 \rceil$$

$$H_{max} = \left\lfloor \log_{\lceil \frac{p}{2} \rceil} \frac{n+1}{2} \right\rfloor$$

unacademy

# Deletion in B-Tree

2 Cases:

1. Deletion in leaf
2. Deletion in internal node

# Deletion in B-Tree: Deletion in Leaf

1. After deletion if no violation of min keys, then no changes in tree
2. If violation of min keys, then borrow key from sibling (rotation through parent).
3. If borrow from sibling can’t be possible then merge the node with sibling and pull down the anchor key from parent.

# Deletion in B-Tree: Deletion in Internal Node

1. Replace the deleted value with inorder successor or inorder predecessor
2. Now follow the rule of deletion of key from leaf node

unacademy

# B+-Tree

| Internal Node | Leaf Node |
| --- | --- |
| • Keys<br>• Tree Pointer | • Keys<br>• Record Pointer |

unacademy

# B+Tree

Order for Internal nodes (not root)

- Every internal node other than root should have atleast $\left\lceil \frac{p}{2} - 1 \right\rceil$ keys or $\left\lceil \frac{p}{2} \right\rceil$ pointers
- Every internal node can have maximum $p - 1$ keys or p pointers
- Every leaf node should have atleast $\left\lceil \frac{q}{2} \right\rceil$ keys and max q keys
- All leaves are on same level
- The leaves are connected using linked list (singly or Doubly)

unacademy

# B+Tree

What if order-4 B+ tree given in question?

unacademy

# Insertion in B+ Tree

Internal nodes order-3

Leaf nodes order-2

Insert 1, 2, 3, 4, 5

Using Node Splitting

unacademy

# Insertion in B+ Tree

Order-5

10, 14,  1, 18,  27, 39, 49, 12, 19, 21, 70, 64, 89, 75

unacademy

# Insertion in B+ Tree

Using Key Distribution

unacademy

# Deletion in B+ Tree

1. After deletion if no violation of min keys, then no changes in tree
2. If violation of min keys, then borrow key from sibling.
3. If borrow from sibling can’t be possible then merge the node with sibling. Either update the anchor key or pull down the anchor key from parent.

unacademy

# Happy Learning.!

unacademy

▲ 1 • Asked by Kumar

Please help me with this doubt

[GATE-1998 : 2 Marks]

18. Given the programming constructs

(i) assignment

(ii) for loops where the loop parameter cannot be changed within the loop

(iii) if-then-else

(iv) forward go to

(v) arbitrary go to

(vi) non-recursive procedure call

(vii) recursive procedure/function call

(viii) repeat loop,

which constructs will you not include in a programming language such that it should be possible to program that terminates (i.e., halting) function in the same programming language.

(a) (ii), (iii), (iv)

(b) (v), (vii), (viii)

(c) (vi), (vii), (viii)

(d) (iii), (vii), (viii)

[GATE-1999 : 2 Marks]