# B-Trees

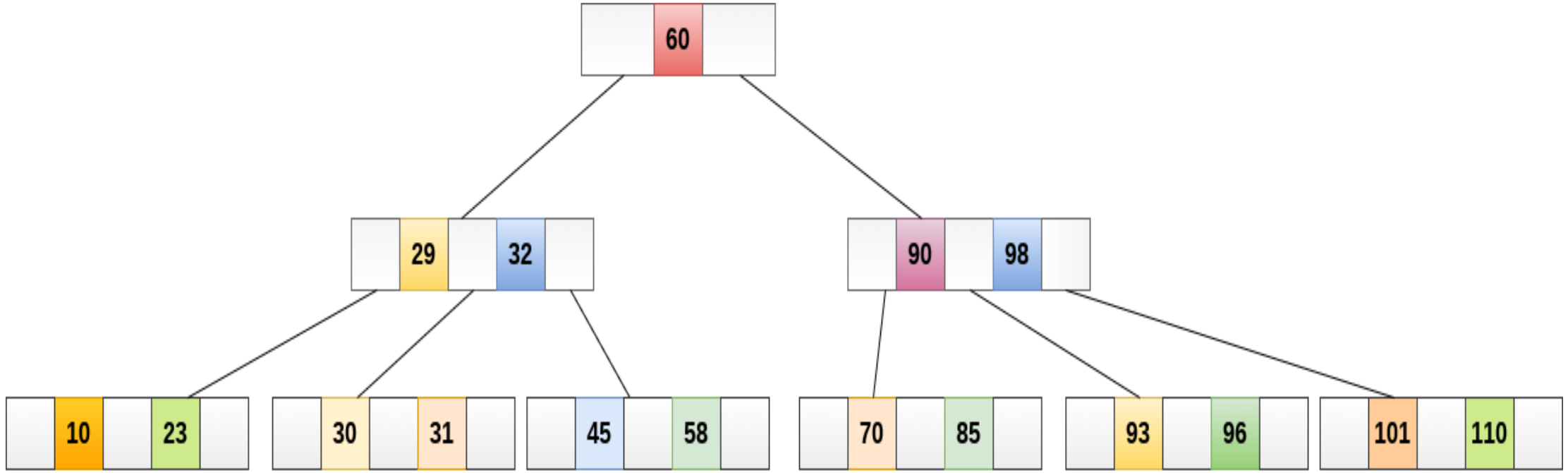
By: Aditya Tiwari
Assistant Professor
CSVTU, Bhilai

# Introduction of B-Tree

- B Tree is a specialized m-way tree that can be widely used for disk access. A B-Tree of order m can have at most m-1 keys and m children. One of the main reasons of using B tree is its capability to store large number of keys in a single node and large key values by keeping the height of the tree relatively small.
- A B tree of order m contains all the properties of an M way tree. In addition, it contains the following properties.

1. Every node in a B-Tree contains at most m children.
2. Every node in a B-Tree except the root node and the leaf node contain at least m/2 children.
3. The root nodes must have at least 2 child nodes.
4. All leaf nodes must be at the same level.

- It is not necessary that, all the nodes contain the same number of children but, each node must have m/2 number of nodes.
- A B tree of order 4 is shown in the following image.

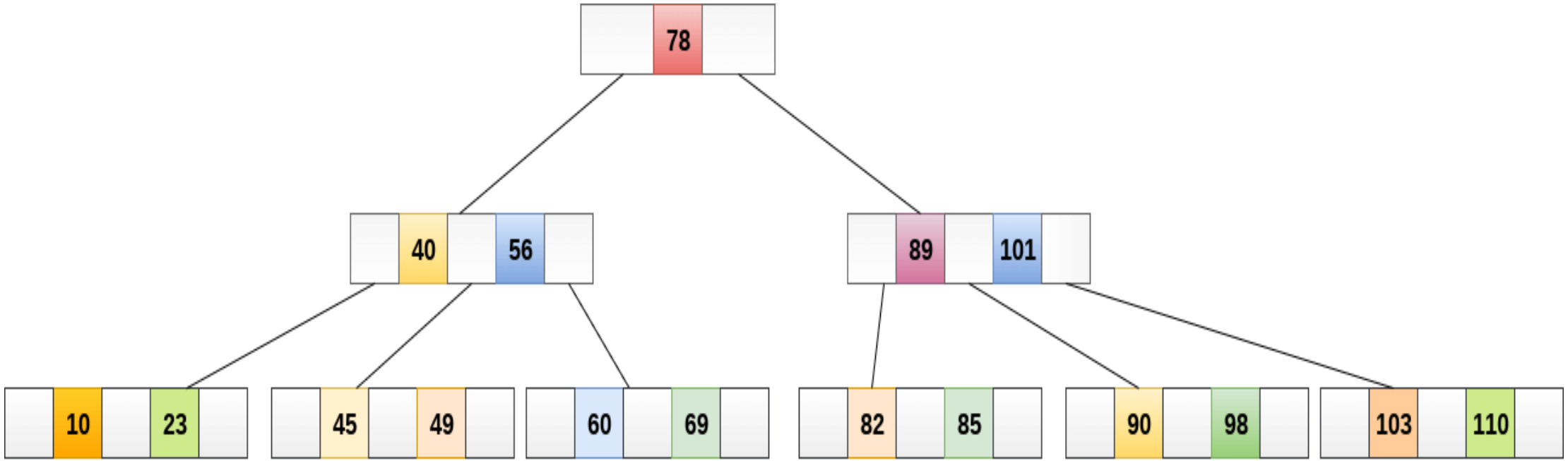

60
29
32
90
98
10
23
30
31
45
58
70
85
93
96
101
110

- While performing some operations on B Tree, any property of B Tree may violate such as number of minimum children a node can have. To maintain the properties of B Tree, the tree may split or join.
- Operations
  - Searching
  - Insertion
  - Deletion

- Searching:
- Searching in B Trees is similar to that in Binary search tree. For example, if we search for an item 49 in the following B Tree. The process will something like following :

1. Compare item 49 with root node 78. since 49 < 78 hence, move to its left sub-tree.
2. Since, 40<49<56, traverse right sub-tree of 40.
3. 49>45, move to right. Compare 49.
4. Match found, return.

- Searching in a B tree depends upon the height of the tree. The search algorithm takes O(log n) time to search any element in a B tree.


78
40
56
89
101
10
23
45
49
60
69
82
85
90
98
103
110

# •Inserting

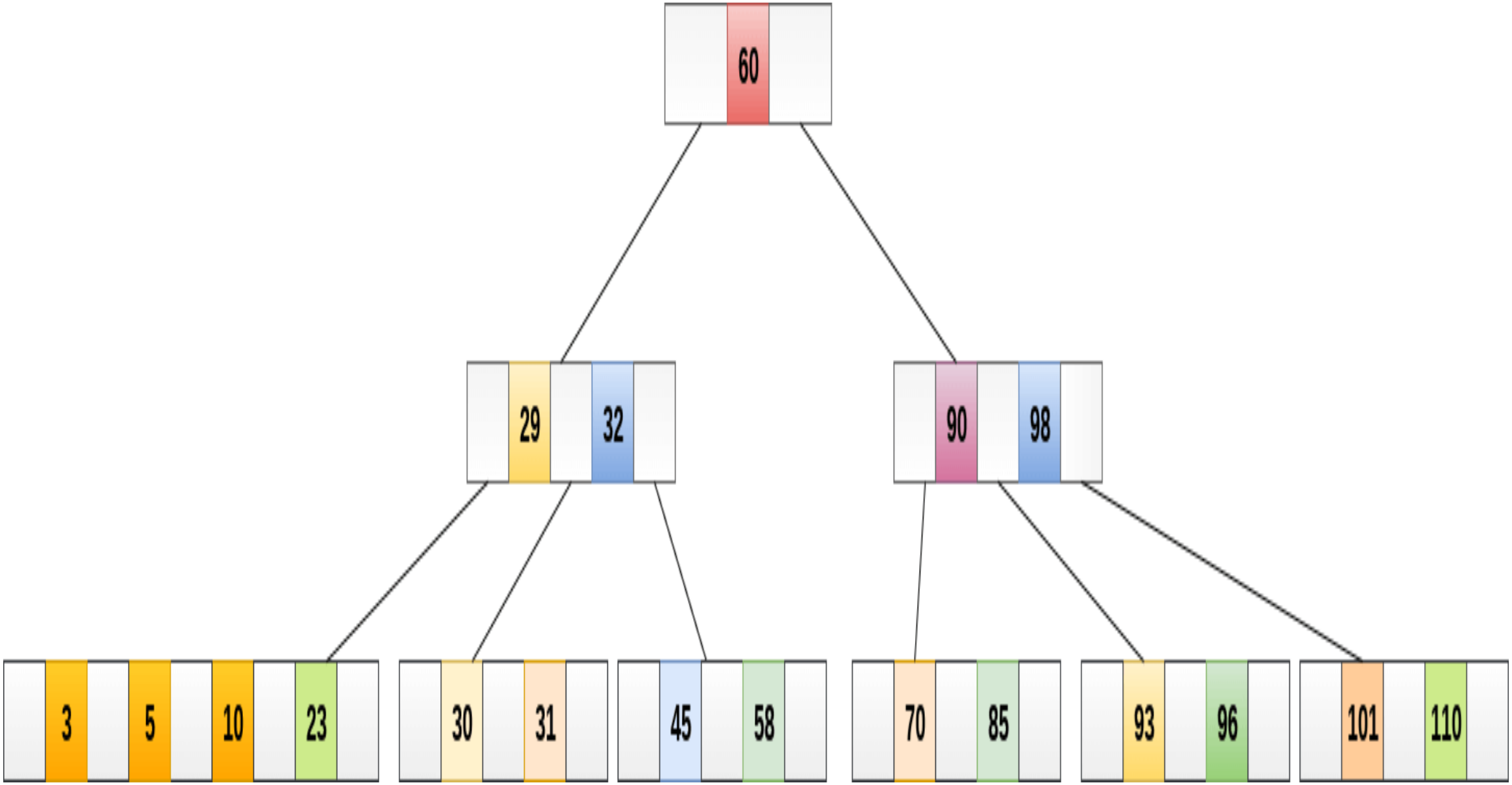
- Insertions are done at the leaf node level. The following algorithm needs to be followed in order to insert an item into B Tree.

1. Traverse the B Tree in order to find the appropriate leaf node at which the node can be inserted.
2. If the leaf node contain less than m-1 keys then insert the element in the increasing order.

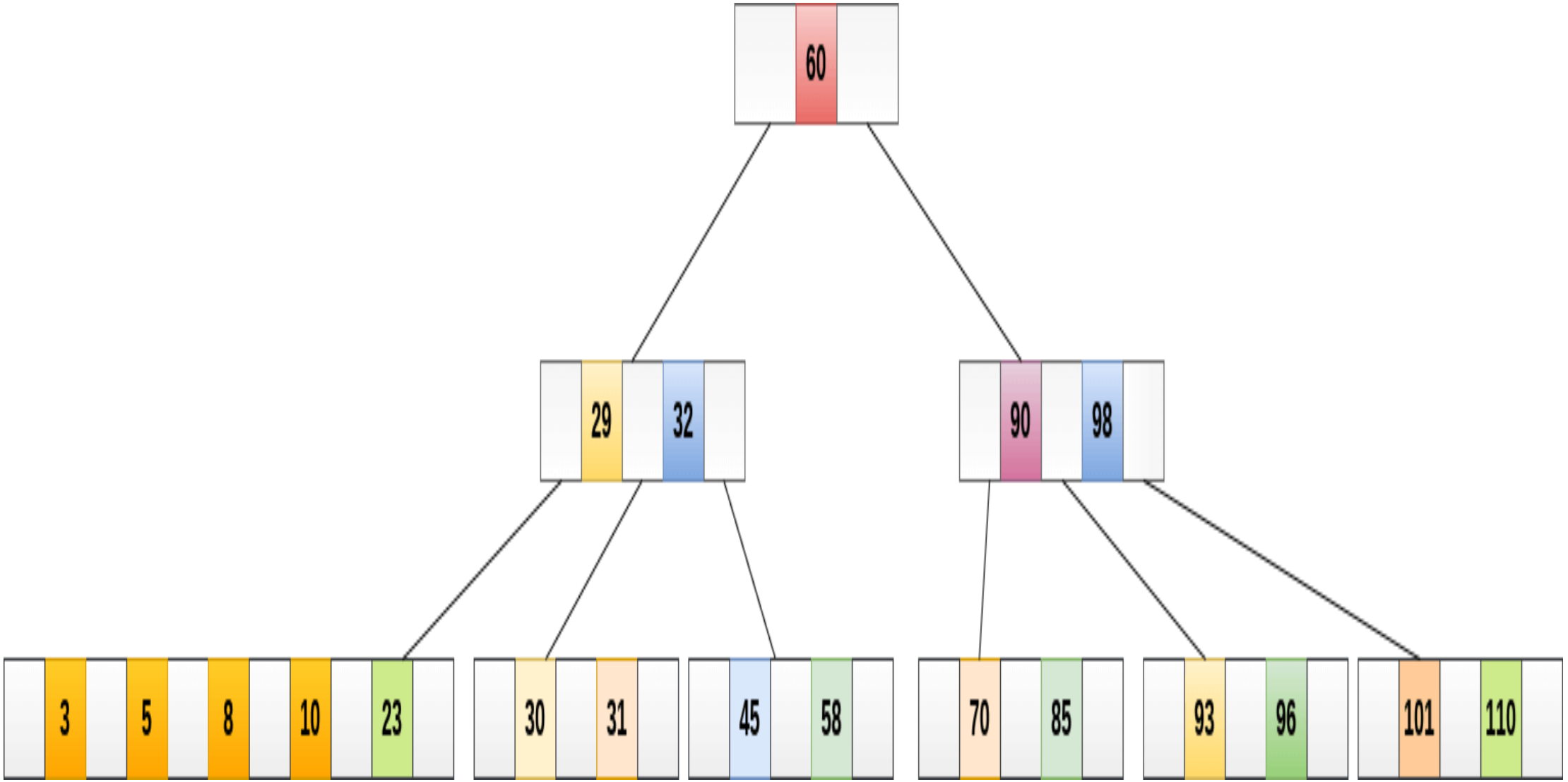
3. Else, if the leaf node contains m-1 keys, then follow the following steps.

- Insert the new element in the increasing order of elements.
- Split the node into the two nodes at the median.
- Push the median element upt o its parent node.
- If the parent node also contain m-1 number of keys, then split it too by following the same steps.

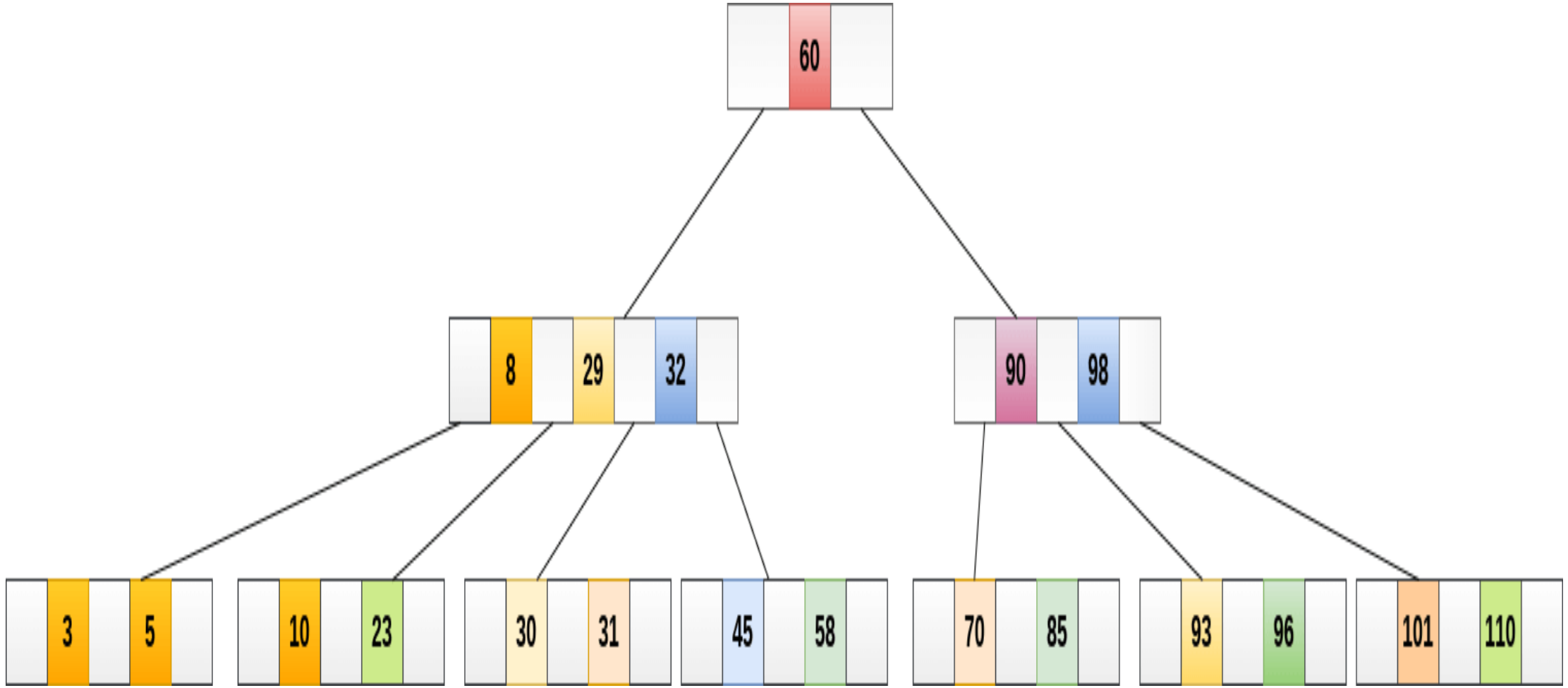
- **Example:**
- Insert the node 8 into the B Tree of order 5 shown in the following image.


60
29
32
90
98
3
5
10
23
30
31
45
58
70
85
93
96
101
110

8 will be inserted to the right of 5, therefore insert 8.



The node, now contain 5 keys which is greater than (5 -1 = 4 ) keys. Therefore split the node from the median i.e. 8 and push it up to its parent node shown as follows.



- THANK YOU