

Book

**A Simplified Approach
to**

Data Structures

Prof.(Dr.) Vishal Goyal, Professor, Punjabi University Patiala

Dr. Lalit Goyal, Associate Professor, DAV College, Jalandhar

Mr. Pawan Kumar, Assistant Professor, DAV College, Bhatinda

Shroff Publications and Distributors

Edition 2014

B TREE

Submitted to – Dr. Vishal goyal

Submitted by – Raman Gaur 140454242
Jagdeep Singh 140454212
Jatinder Singh 140454166

Date of submission – 30 March, 2015

B-tree

A very important category of m-way trees is b-tree which was introduced by **R.Bayer** and **E.Mcgreight**

B-tree of order m has following properties:

- Each node of tree except root node must have maximum of **m-1** keys and minimum of **m/2-1** keys.
- All leaf nodes in B-tree must be on same level.
- Each node of tree except root node and leaf nodes has maximum of **m** children and minimum of **m/2** children.
- The key values in each node are stored in ascending order.
- The key in node separates the ranges of keys which are atored in each child of that node.

A B-tree of order m and height h has total number of elements **=m^(h+1)-1**.

Operations performed on B-tree.

1. Searching
2. Insertion
3. Deletion

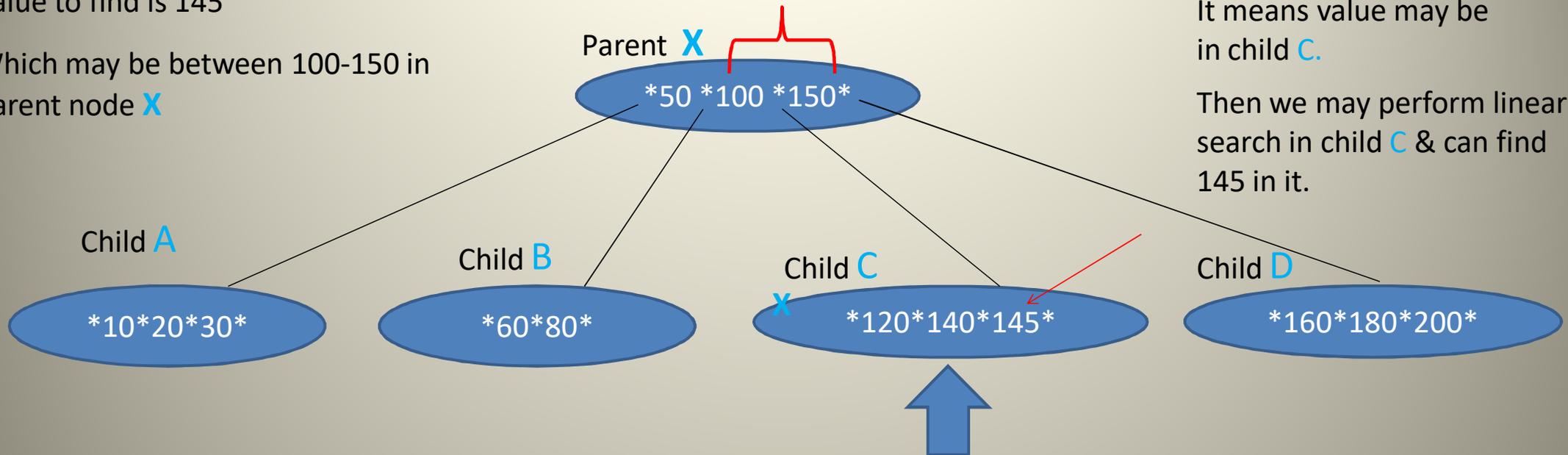
- **Searching:**

if the B-tree is of order 4 then we have maximum of 4 children of each node and hence 4way choice to move to one of 4-children.the correct child is chosen by performing a linear search of keys in a node.

SEARCHING IN B TREE

Value to find is 145

Which may be between 100-150 in parent node X



It means value may be in child C.

Then we may perform linear search in child C & can find 145 in it.

For example, consider a B tree of order 4.

Suppose we want to search 145 then first of all we go to root node and find that it is between 100-150, so we go to third child of this root node. In this node we again perform linear search. This time we find the desired key value in this node

Example of Insertion in B-Tree(1)

Insert <10, 50, 30>



Sort

Insert <70, 90>



Overflow

Insert 25

Split
Promotion



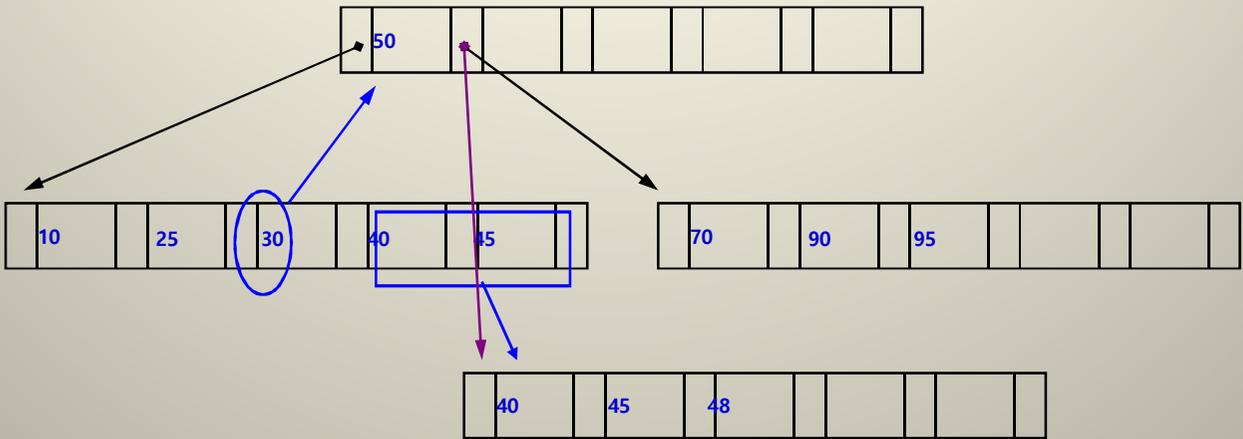
Insert <40, 95, 45>



Example of Insertion in B-Tree(2)

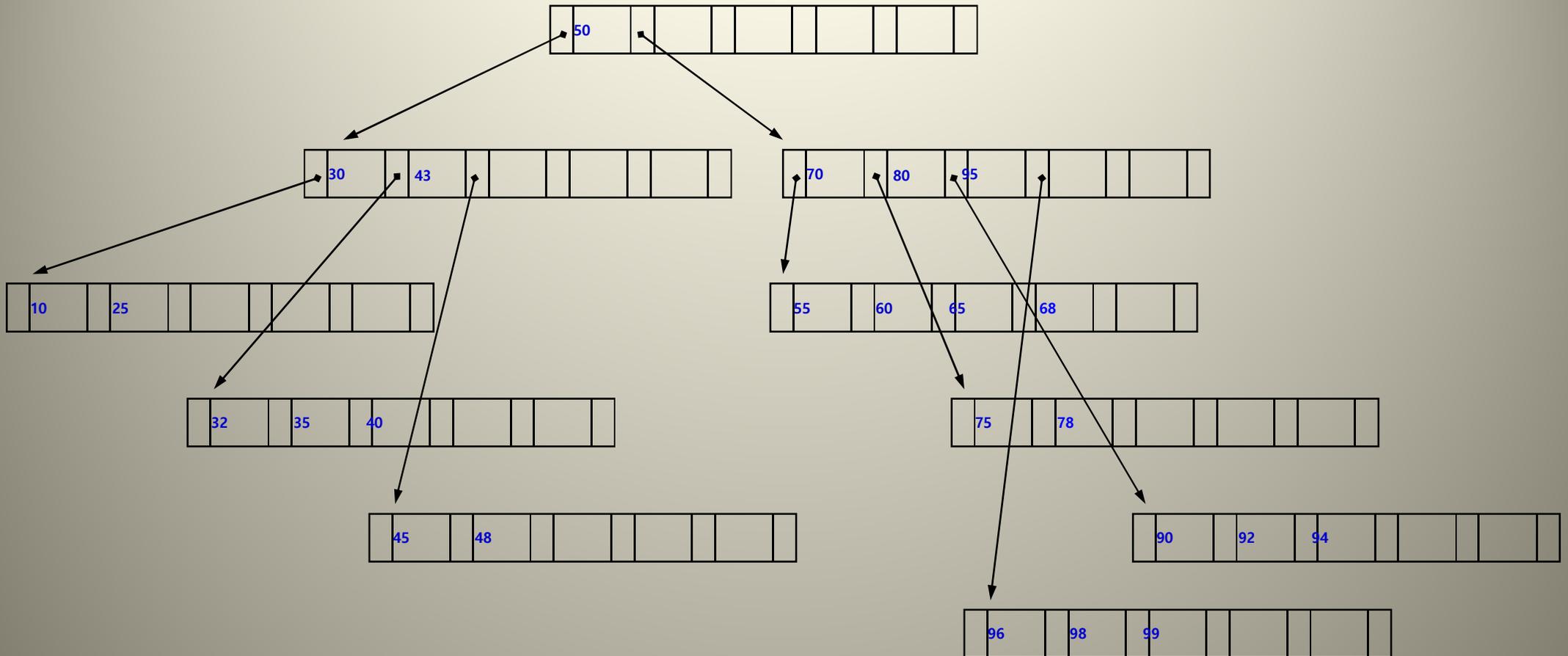
Insert 48

Overflow

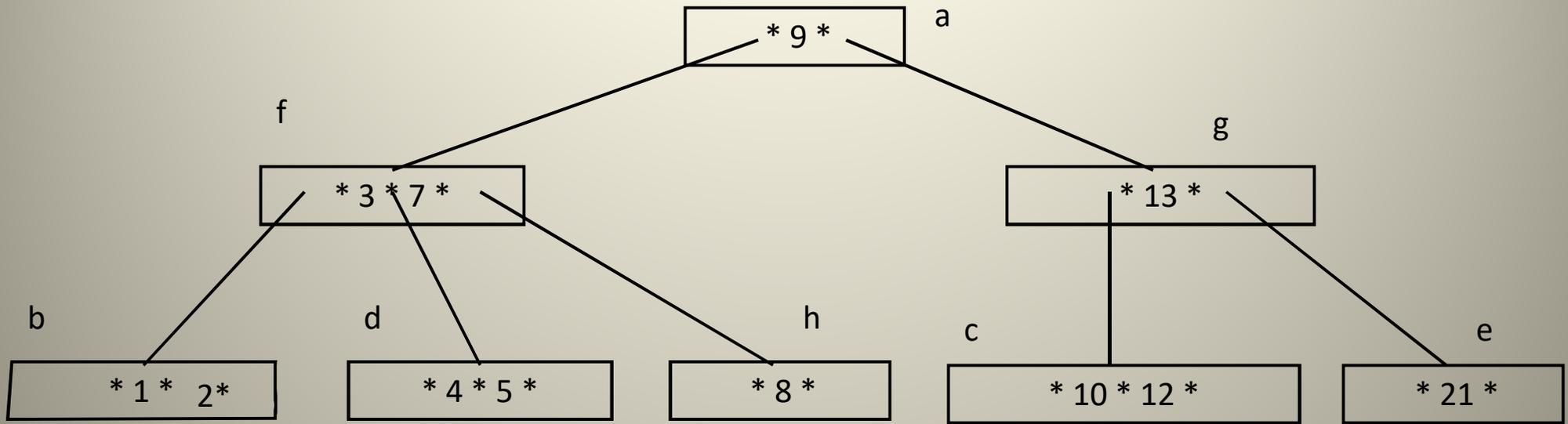


Insert <60, 80, 35, 75, 43, 32, 78, 65, 55, 68, 99, 92, 96, 98, 94>

Example of Insertion in B-Tree(3)

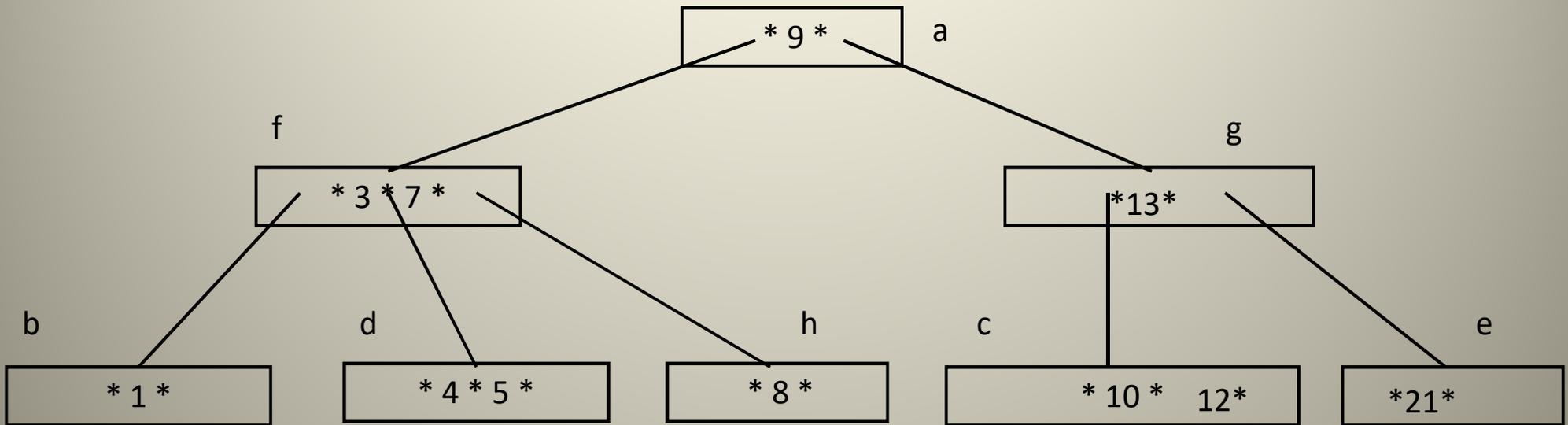


Delete 2

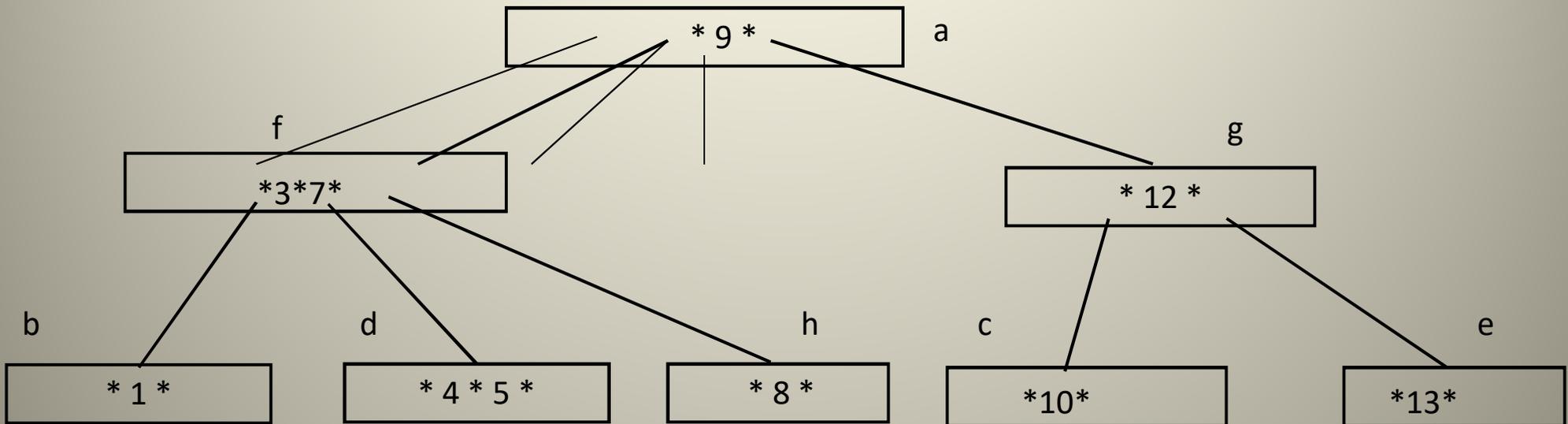


Now, we delete **2**

Delete 21

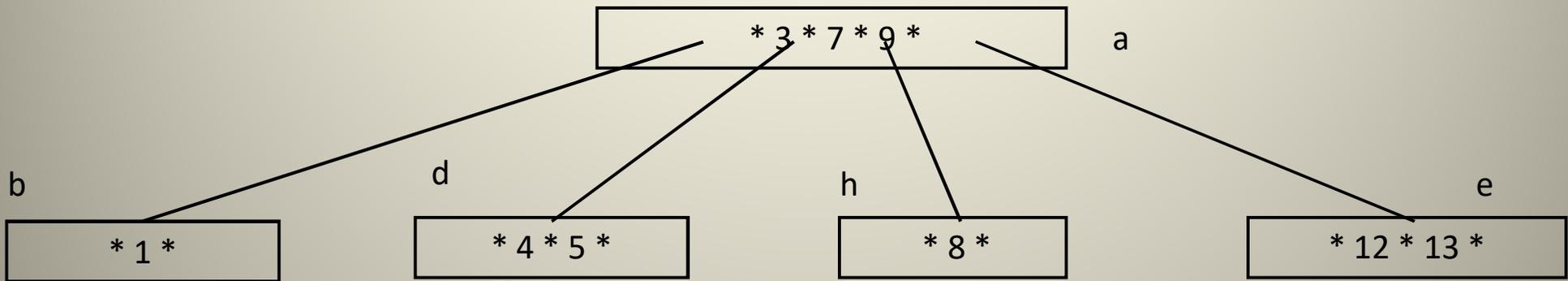


Delete 10



Deleting 10 causes node c to underflow. This causes the parent, node g to recombine with nodes f and a. This causes the tree to shrink one level.

Delete 10



Deleting 10 causes node c to underflow. This causes the parent, node g to recombine with nodes f and a. This causes the tree to shrink one level.