جامعة بيرزيت
BIRZEIT UNIVERSITY

B-Tree

Dr. Abdallah Karakra

Computer Science Department

COMP242

Monday, June 5, 2023

B-Tree

- A B-Tree is a multi-way search tree of nodes "m" such that the following properties hold:
  - The **root** is either a leaf or has between 2 and m children.
  - If a node has 't' number of children then it must have (t-1) number of keys.
  - The keys of a node 'x' are sorted in non-decreasing orders.

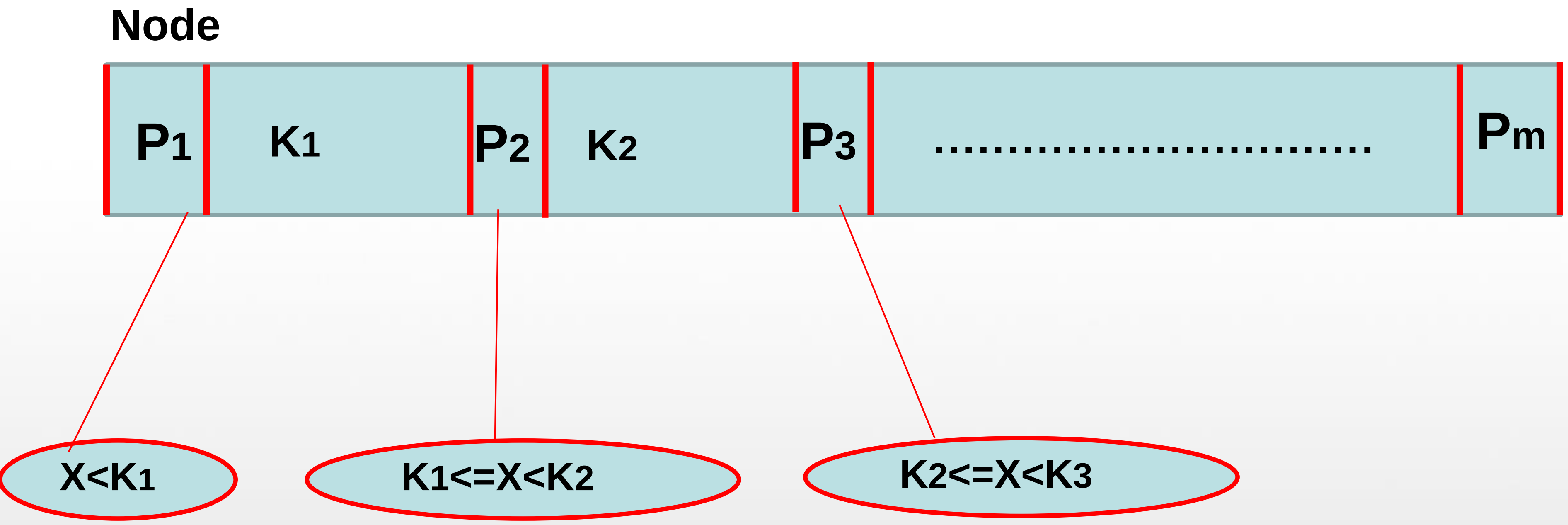
B-Tree

- A B-Tree is a multi-way search tree of nodes “m” such that the following properties hold:
  - The root may have at most ‘m’ non-empty children if the root is not itself a leaf node. If the root is also a leaf node then it may not have any child.
  - All leaf nodes are on the same level, which defines the height of the tree.
  - All data records are stored at the leaves
  - **Internal nodes only used for searching**

B-Tree: You have to know!

- 2-3 and 2-3-4 trees are types of a B-tree.
- A B-Tree of order 3 is also called 2-3 tree.
- A B-Tree of order 4 is also called 2-3-4 tree.
- B-Tree is perfectly balanced. find, insert, and remove operations take O(log n) time, even in the worst case.
- 2-3-4 trees are thus named because every node has 2, 3, or 4 children, except leaves, which are all at the bottom level of the tree. Each node stores 1, 2, or 3 entries, which determine how other entries are distributed among its children's subtrees.

Structure of nodes of B-Tree

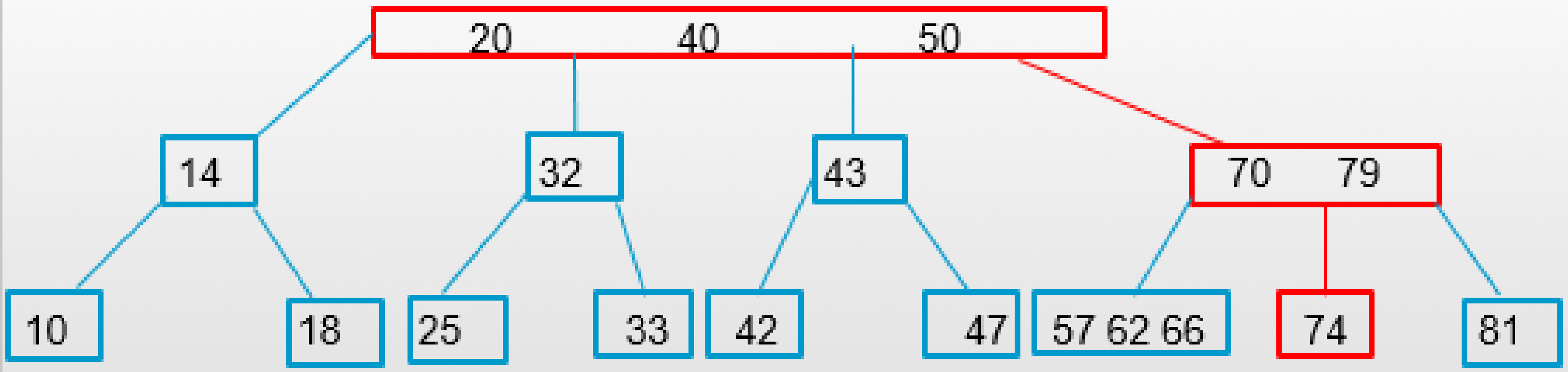


B-Tree: Tree Operations

[1] Object find(Object k);

Finding an entry is straightforward. Start at the root. At each node, check for the key k; if it's not present, move down to the appropriate child chosen by comparing k against the keys. Continue until k is found, or k is not found at a leaf node.

For example,find(74) visits the **red boxes through the red lines** at right.



Example (1)

[2] void insert

Example (1)

In this example:

- **I'll use the * to denote for the pointers.**
- **For more clarification and to make the things easier I'll use the letters a,b,…**

B-Tree of order 4

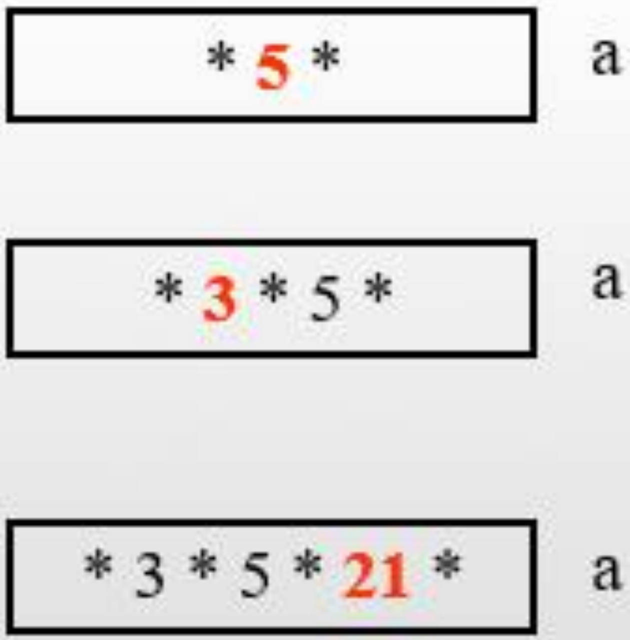
First of all, you have to know:

- → **max no. of child =4**
- → **max no. of key=3**
- → **min no. of child=2**
- → **min no. of key= 1**

Insert: 5, 3, 21, 9, 1, 13, 2, 7, 10, 12, 4, 8

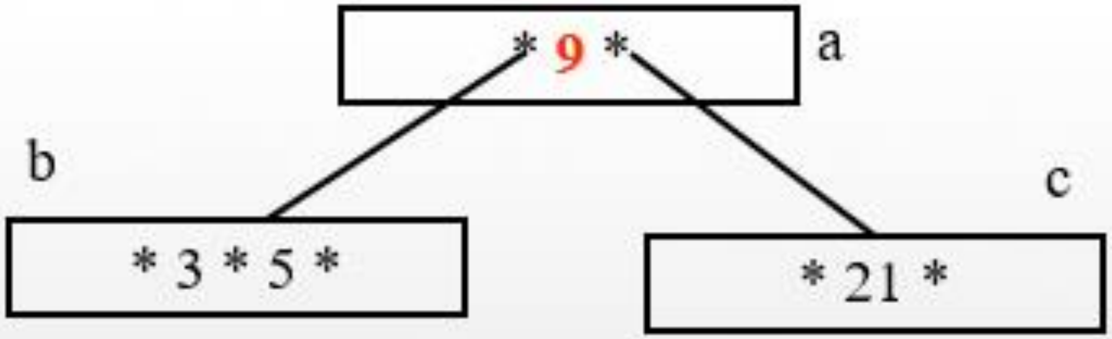
Example (1)

Insert 5, 3, 21



Example (1)

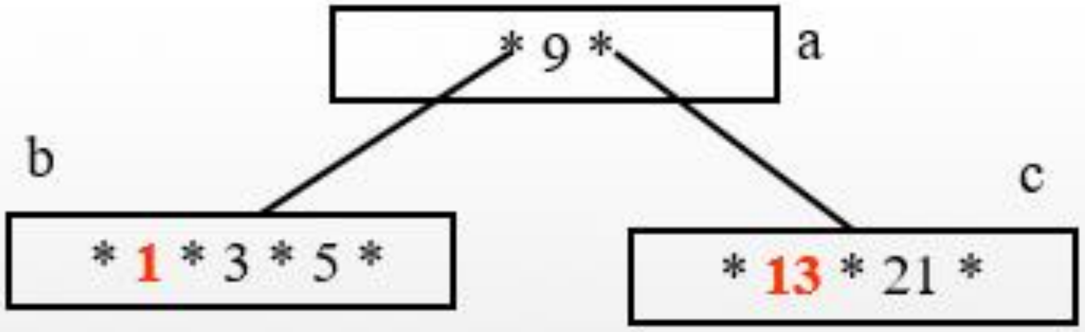
Insert 9



Node a splits creating 2 children: b and c

Example (1)

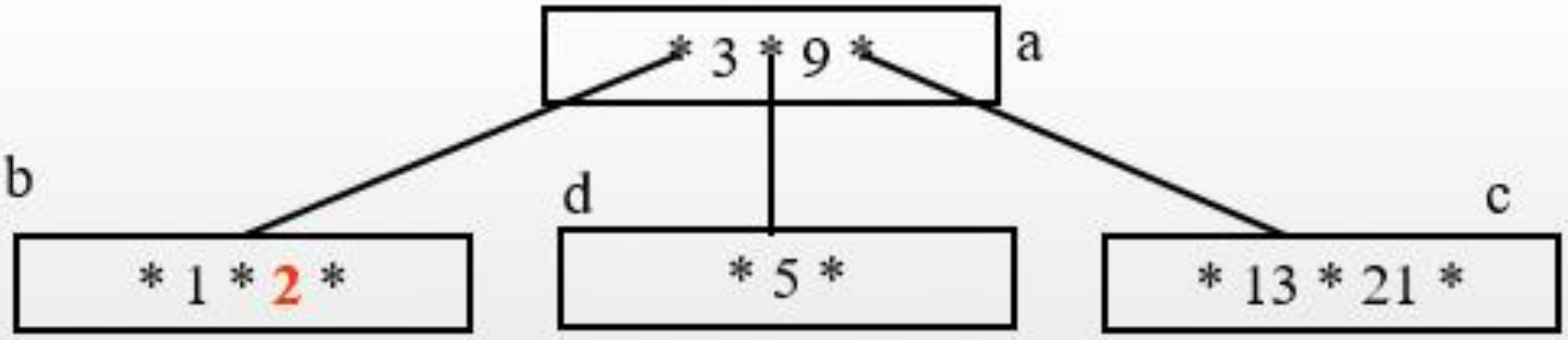
Insert 1, 13



Nodes b and c have room to insert more elements

Example (1)

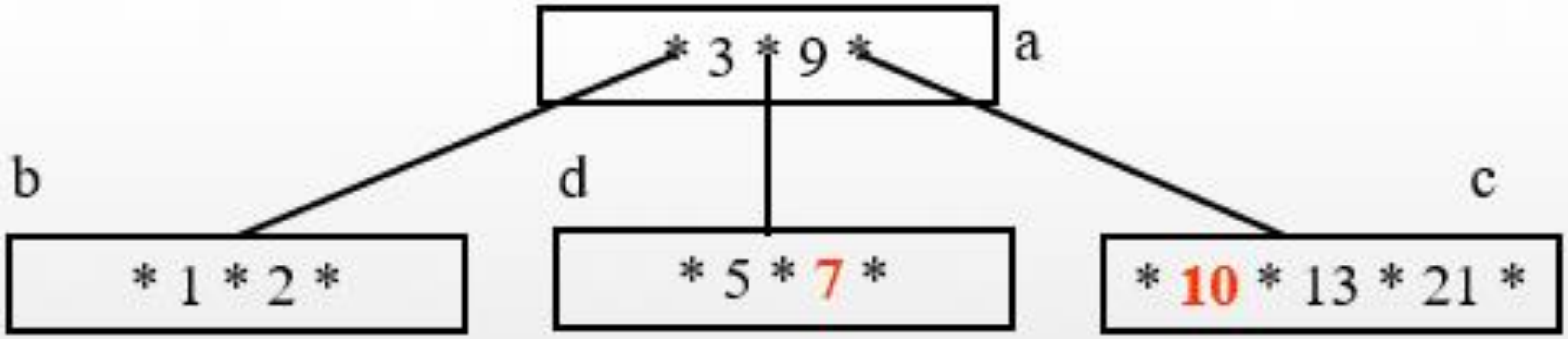
Insert 2



Node b has no more room, so it splits creating node d.

Example (1)

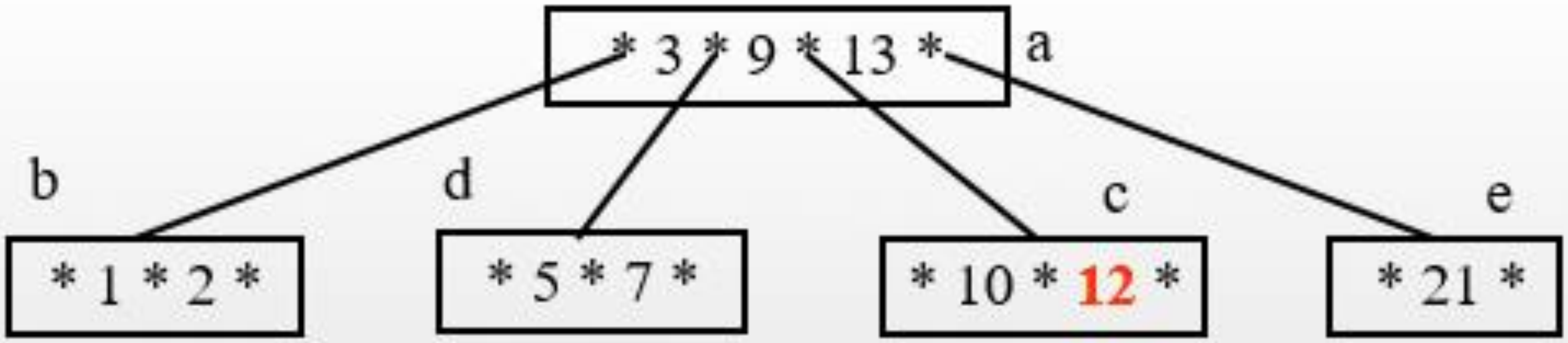
Insert 7, 10



Nodes d and c have room to add more elements

Example (1)

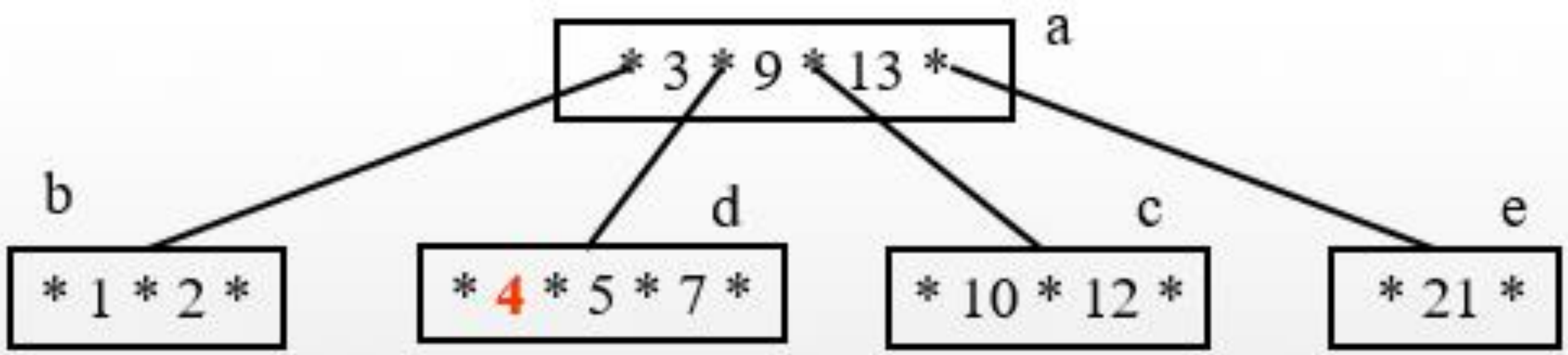
Insert 12



Nodes c must split into nodes c and e

Example (1)

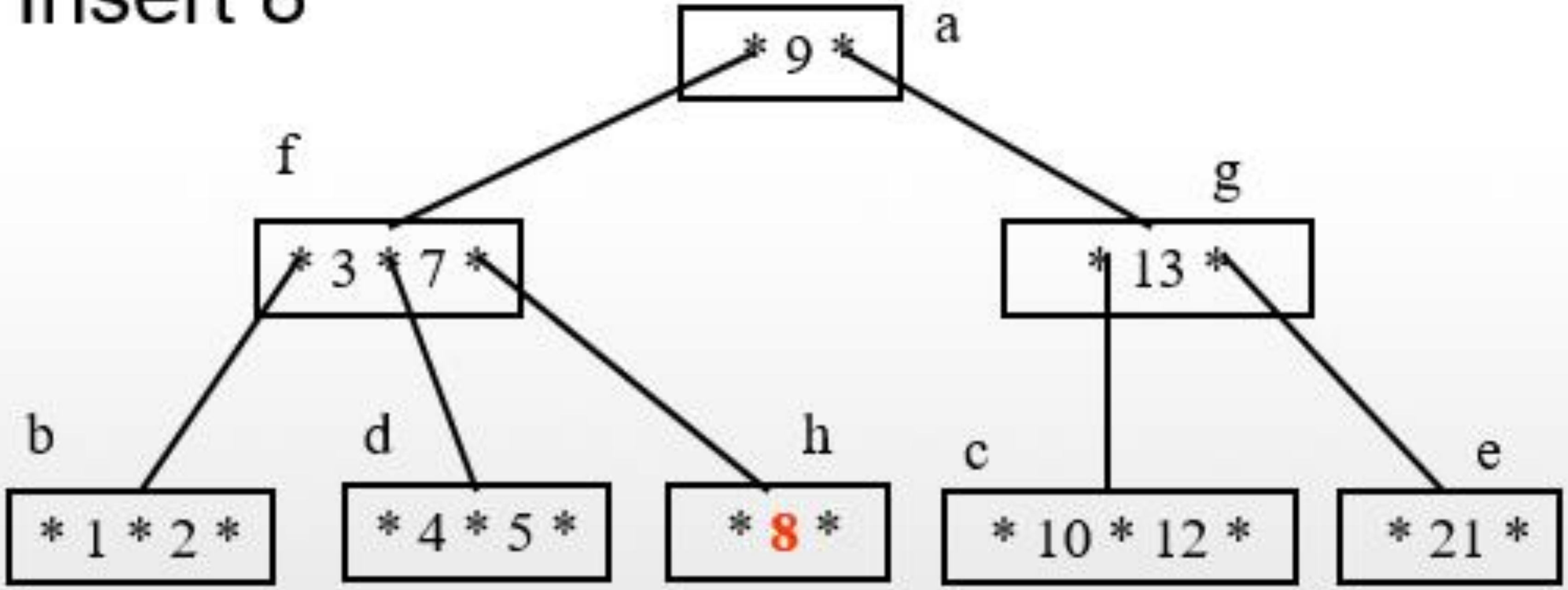
Insert 4



Node d has room for another element

Example (1)

Insert 8



Node d must split into 2 nodes. This causes node a to split into 2 nodes and the tree grows a level.

B-Tree: Tree Operations

[2] void insert function

Example (2)

B-Tree of order 3

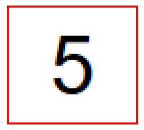
First of all, you have to know:

- → **max no. of child =3**
- → **max no. of key=2**
- → **min no. of child=2**
- → **min no. of key= 1**

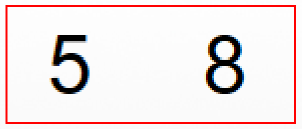
Insert: 5,8,9,20,30,15,16,14,13,31

Example (2)

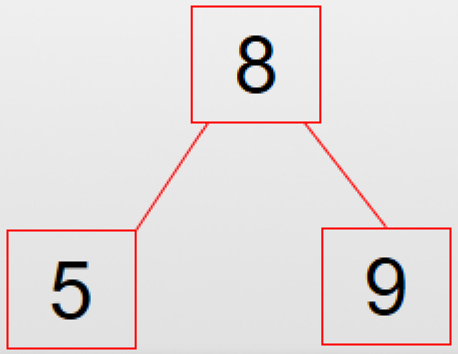
Insert 5



Insert 8

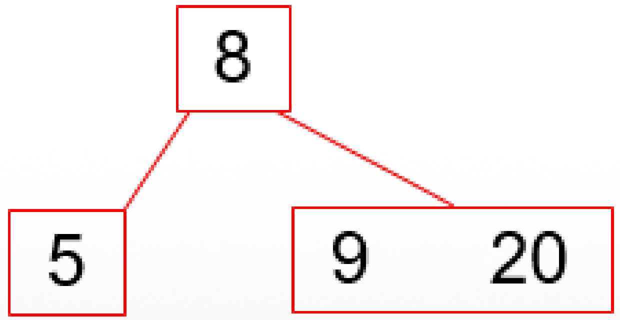


Insert 9

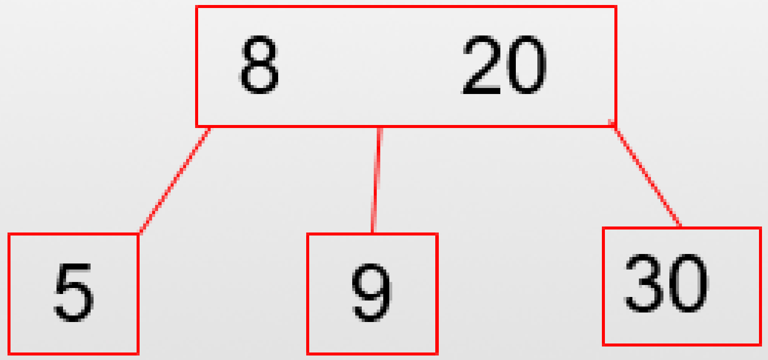


Example (2)

Insert 20

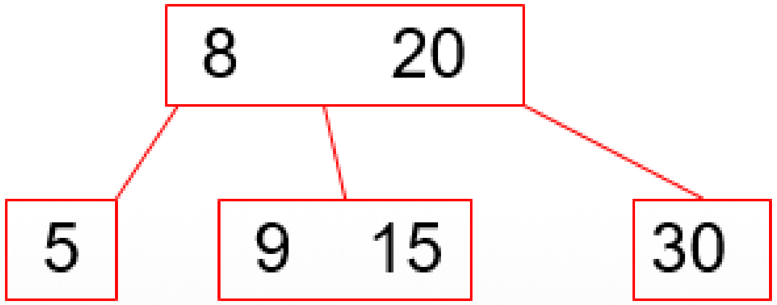


Insert 30

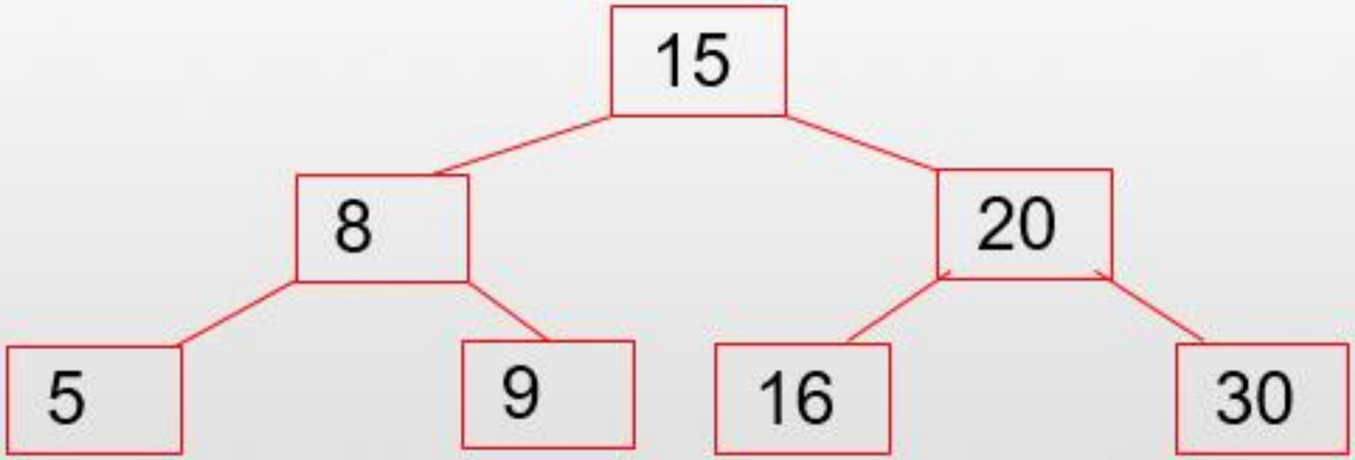


Example (2)

Insert 15

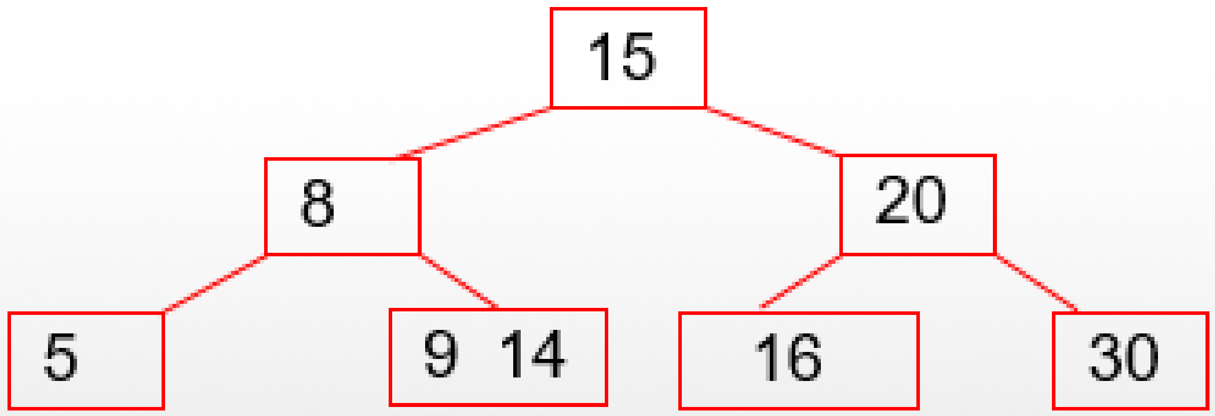


Insert 16



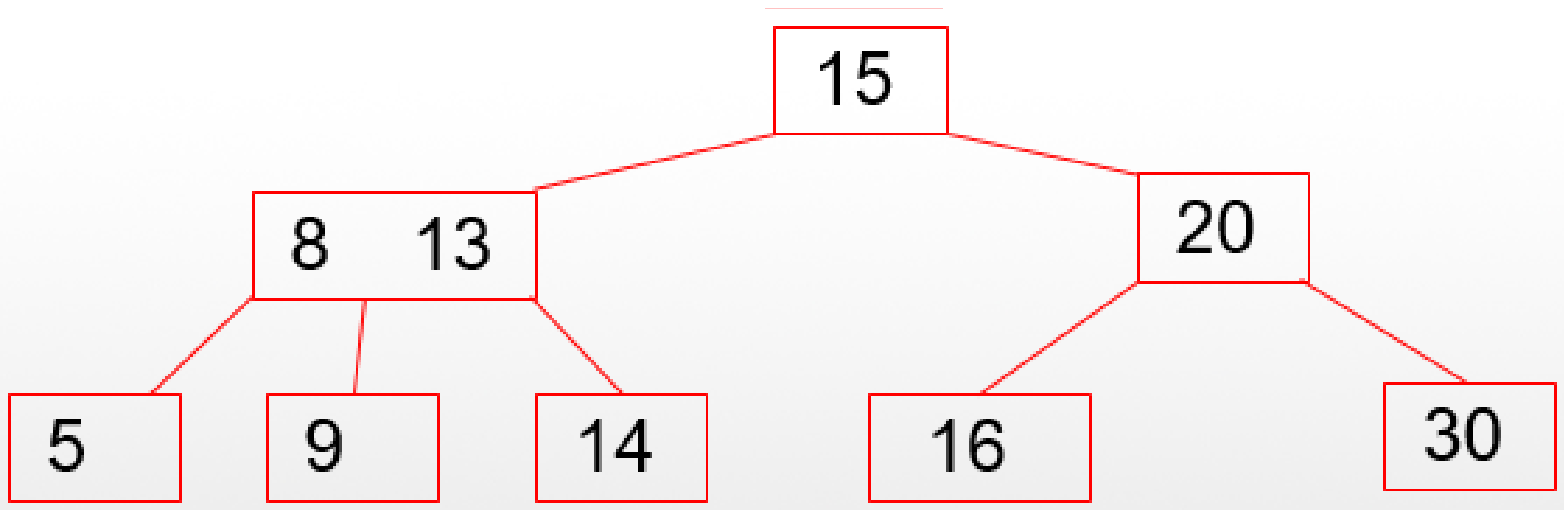
Example (2)

Insert 14



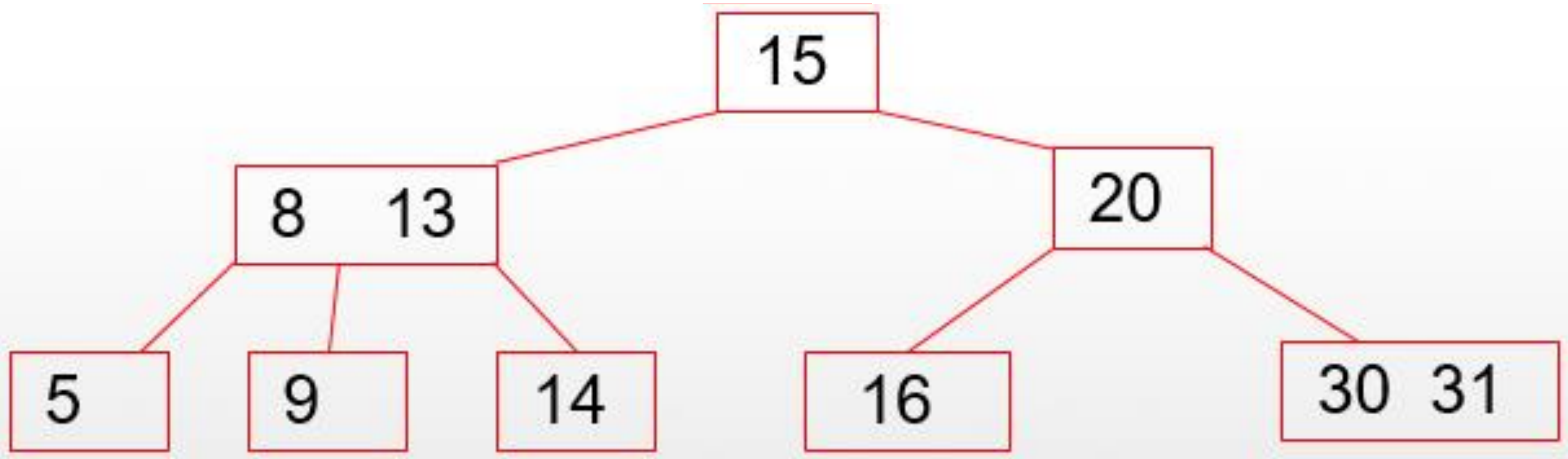
Example (2)

Insert 13



Example (2)

Insert 31



B-Tree: Tree Operations

[3] void remove function

Extra Exercises

1. Insert the following values in this order into a B-tree with M=3

(5,4,7,3,2,1,9,8,6)

2. Insert the following values in this order into a B-tree with M=4

(5,4,7,3,2,1,9,8,6)

Question?

“Success is the sum of small efforts, repeated day in and day out.”
Robert Collier

Reference:

Michael T. Goodrich and Roberto Tamassia, *Data Structures and Algorithms in Java*, John Wiley & Sons, 2010. ISBN # 0-470-38326-7.

GATE and NET Computer Science video-lec

2-3-4 TREES lecture by Prof. Jonathan Shewchuk