BST

CSC-114 Data Structure and Algorithms


Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Outline

- Search Trees
  - Binary Search Tree
    - Search
    - Insertion
    - Deletion
    - Constructing with Traversal Orders

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Binary Search Tree

A very popular search tree which supports efficient processing of data. It stores data in a manner which allows faster lookup.

A Binary Search Tree is a binary tree with a special property:

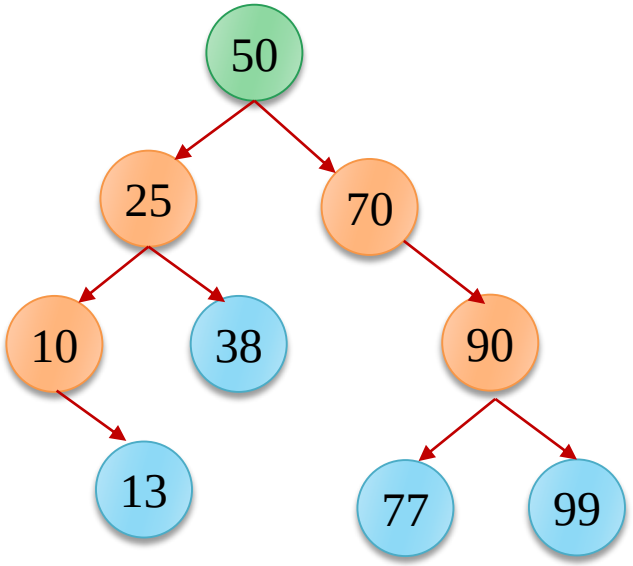
Each node has some compare able data field(key) and it fulfills following rule:

Node's left sub tree holds keys less than the node's key,

Node's right sub tree holds keys greater than the node's key.

Note: No duplicate keys are allowed

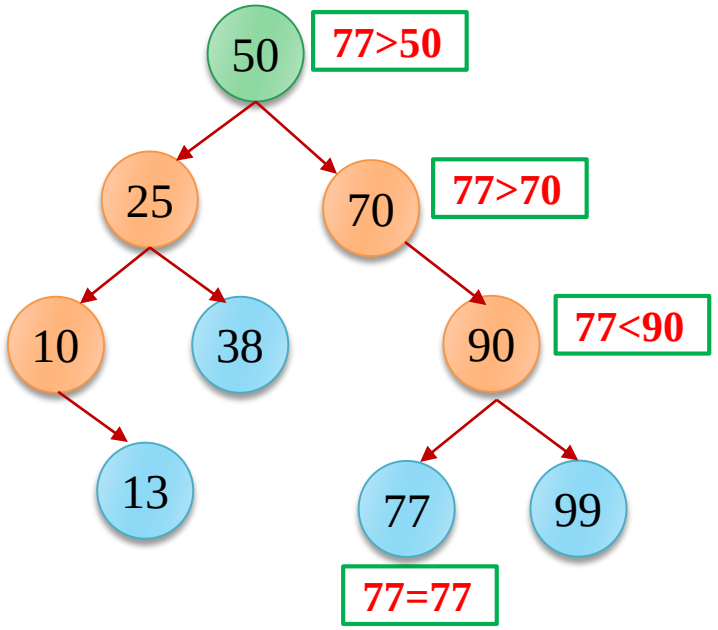
- BSTs are used to present sorted data:
  - If you traverse tree in order, it will produce sorted keys:
    - 10 13 25 38 50 70 77 90 99
  - Operations:
    - Search
    - Insertion
    - Deletion



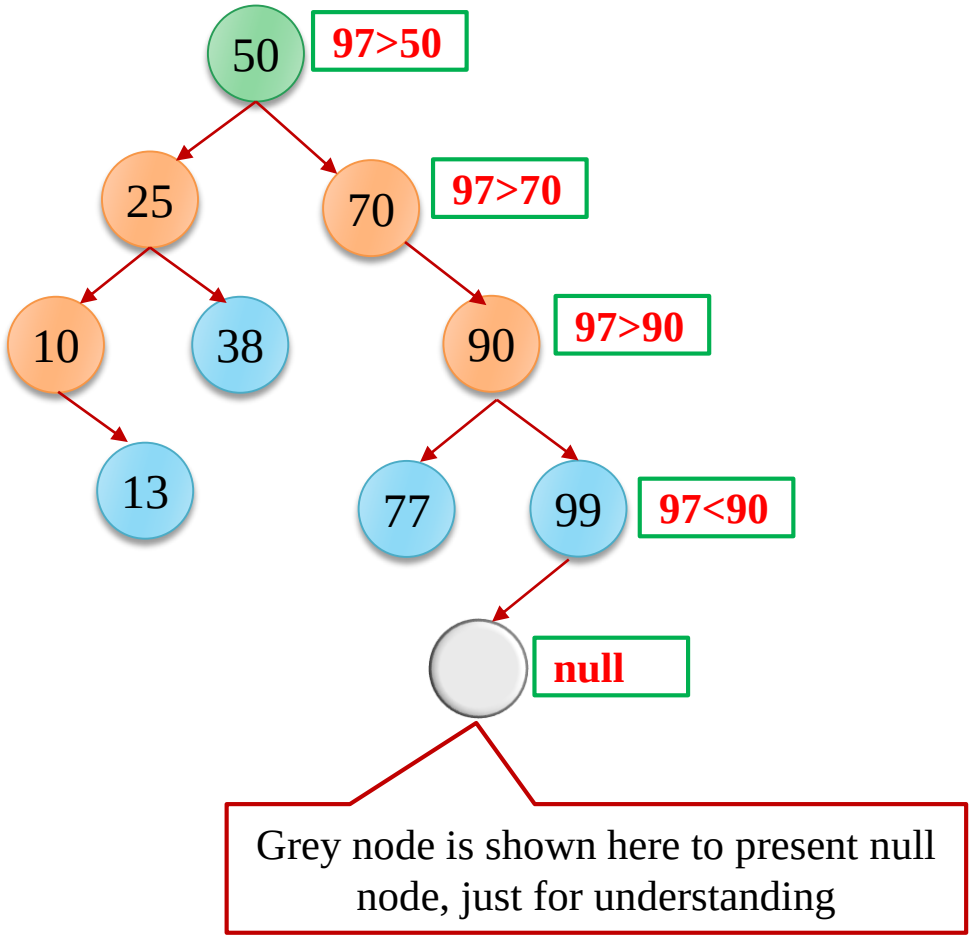
Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Search

Search 77



Search 97



How much time will it take?

What would be the worst case scenario?

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Search: Algorithm

BST_SEARCH(node, key)

Input: root node of tree, key to be searched

Output: node which contains value

Steps:

Start

1. If node== null OR value==node.key //one case for both value found or not found
   return node
2. Else If value < node.key
3. return BST_SEARCH(node.left, key)
4. Else // key> node.key
5. return BST_SEARCH(node.right, key)

End

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Insertion

Let say we want to insert 5

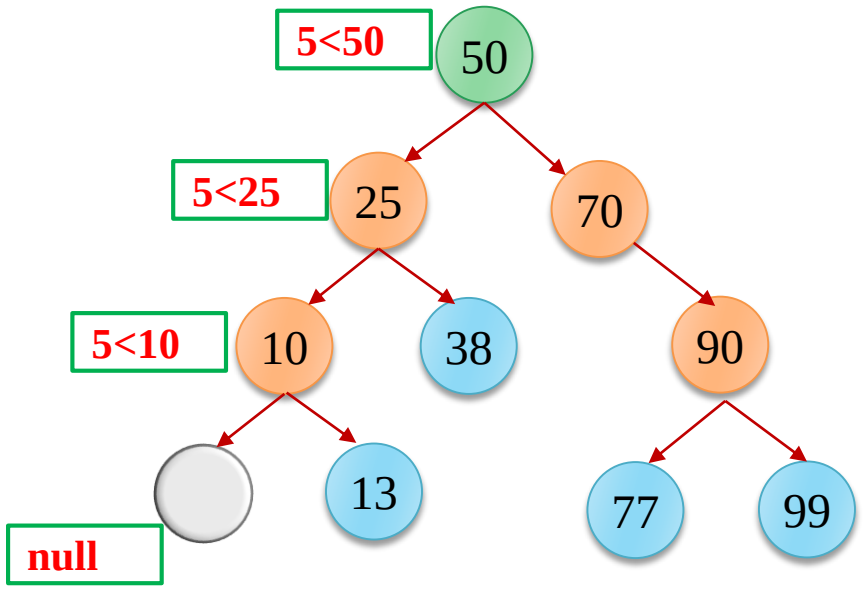
What is the correct position for it

5<10

Move to left

Left child of 10 is null

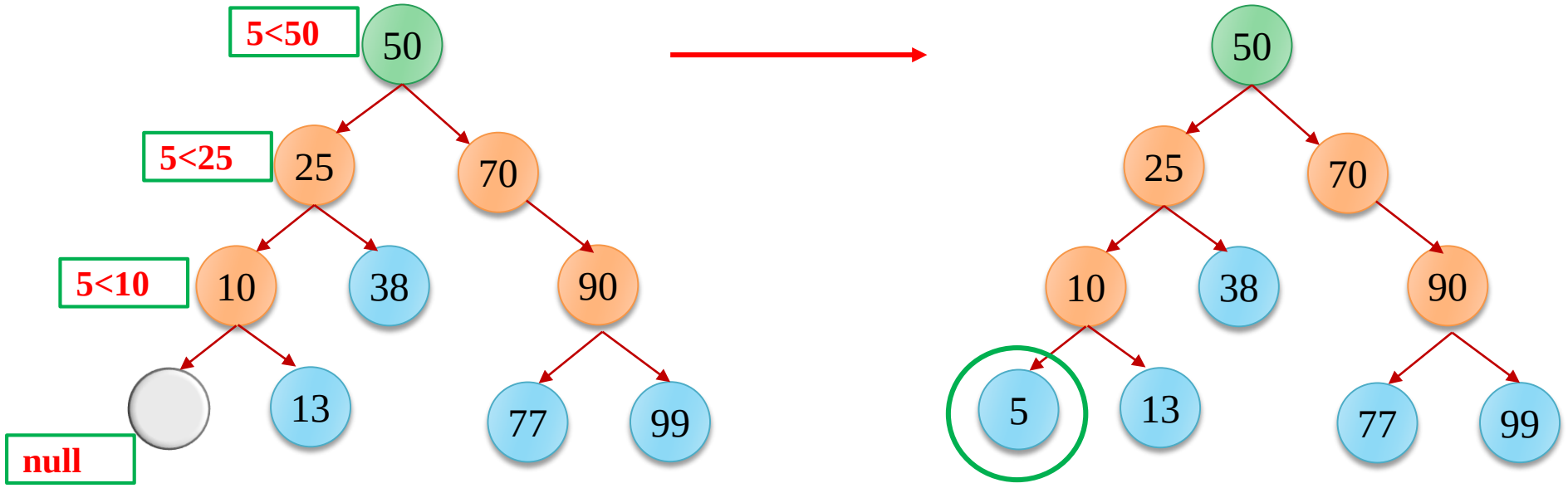
Make 5 it's left child



Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Insertion

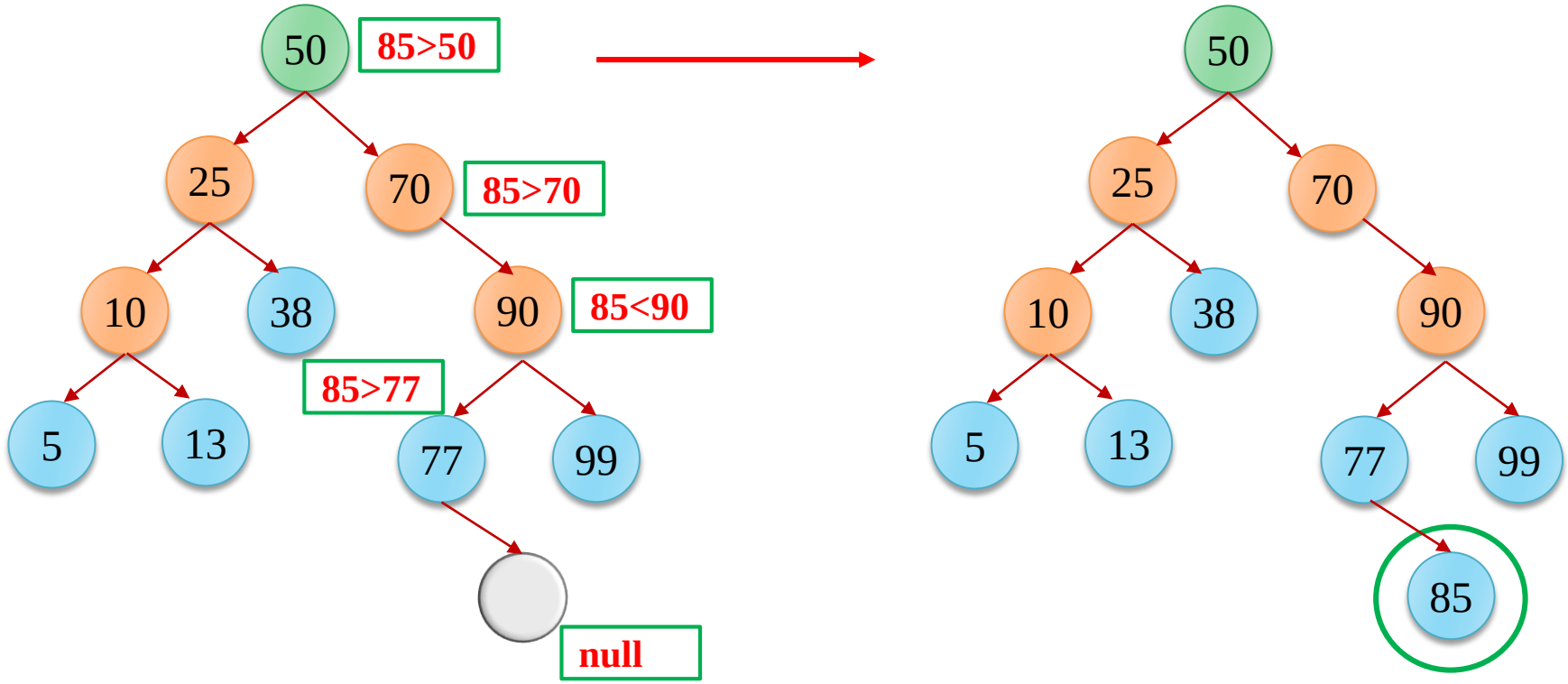
Insert 5



Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Insertion

Insert 85



Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Insert:Algorithm

- Have you observed?
  - New node is always inserted as leaf node
- So
  - If root is null
    - New node is root
  - And if root node is not null
    - Some other node's links will be updated
      - If new node is smaller, update left child
      - Else update right child
  - How to write a recursive function?
    - What it should receive?
    - What is should return?

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Insert:Algorithm

BST_INSERT(node, newNode)

Input: root node of tree, node to be inserted

Output: updated tree with newNode

Steps:

1. If node == null //base case
2. node = newNode
3. Else If newNode.key < node.key
4. node.left = BST_INSERT(node.left, newNode)
5. Else If newNode.key > node.key
6. node.right = BST_INSERT(node.right, newNode)
7. End If
8. Return node

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

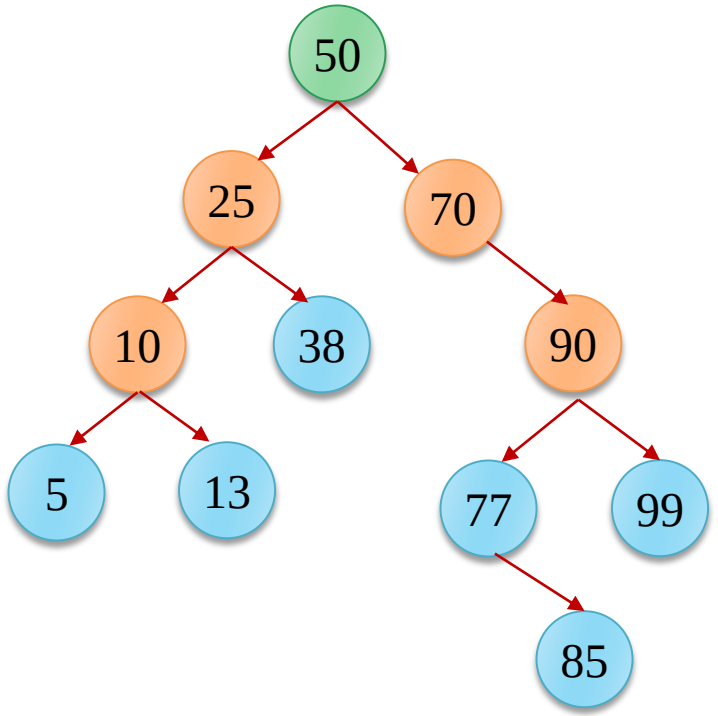
Deletion

After deletion of any node, Tree must remains BST

Let say:

- We want to delete 5
- We want to delete 70
- We want to delete 25

What is difference between these nodes?

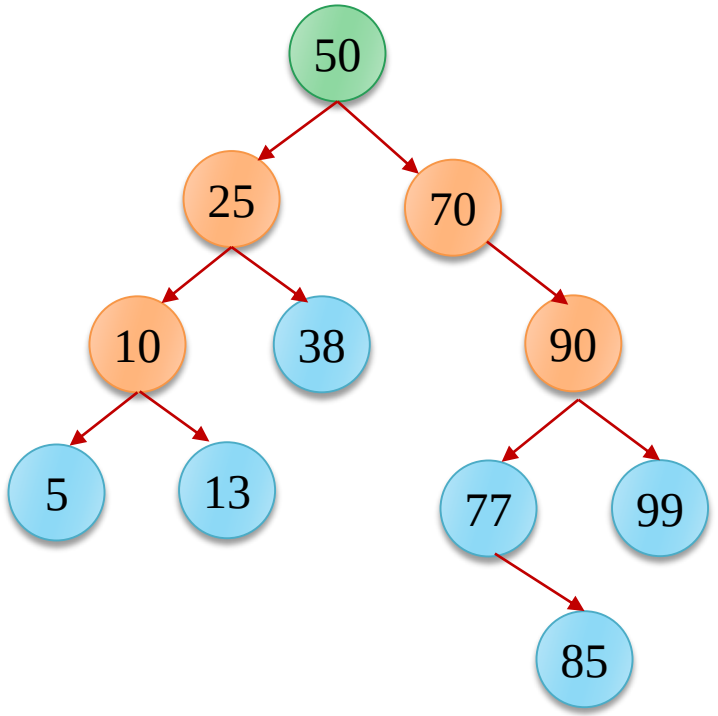


Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

There are three cases when deleting a node:

- Node is leaf node
  - Simply delete
- Node has only one child
  - Link parent of node to child of node
- Node has two child
  - Complex case

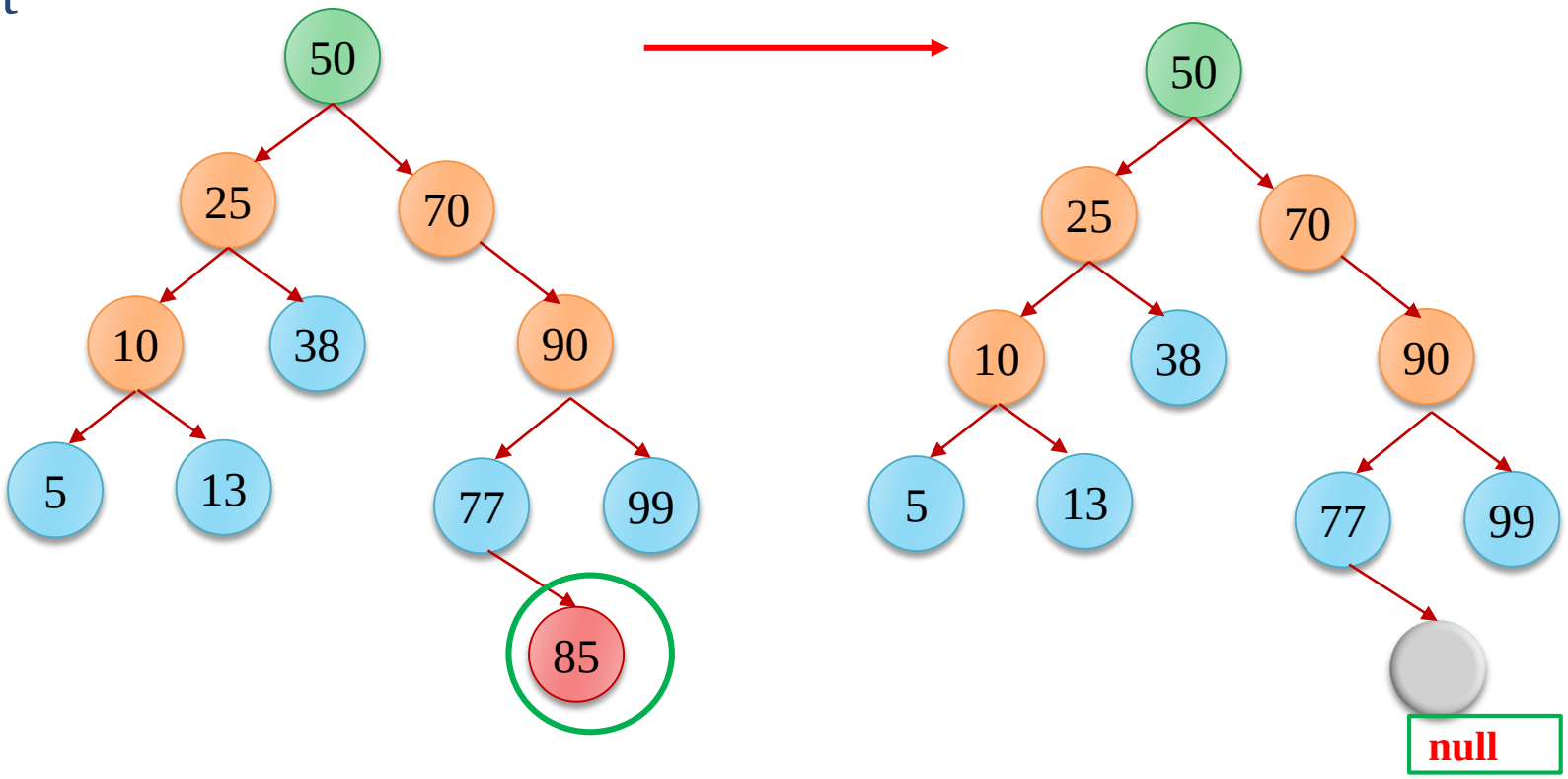


Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 1: If node is leaf node

Unlink from parent

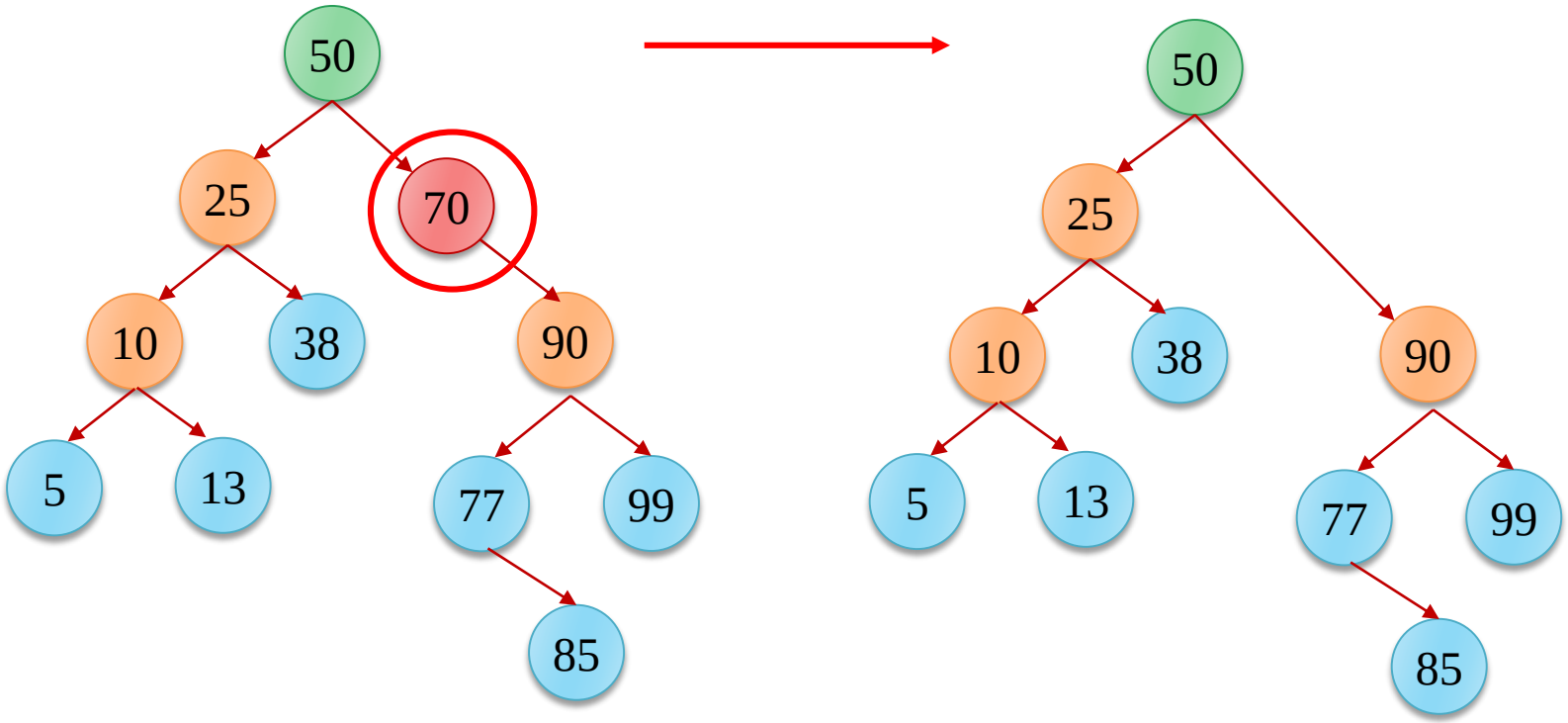


Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 2: If node has one child

Link parent of node to child of node



Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 3: If Node has two child

Consider node 25 in following modified tree

Left link of 50 needs to be updated

But with which node?

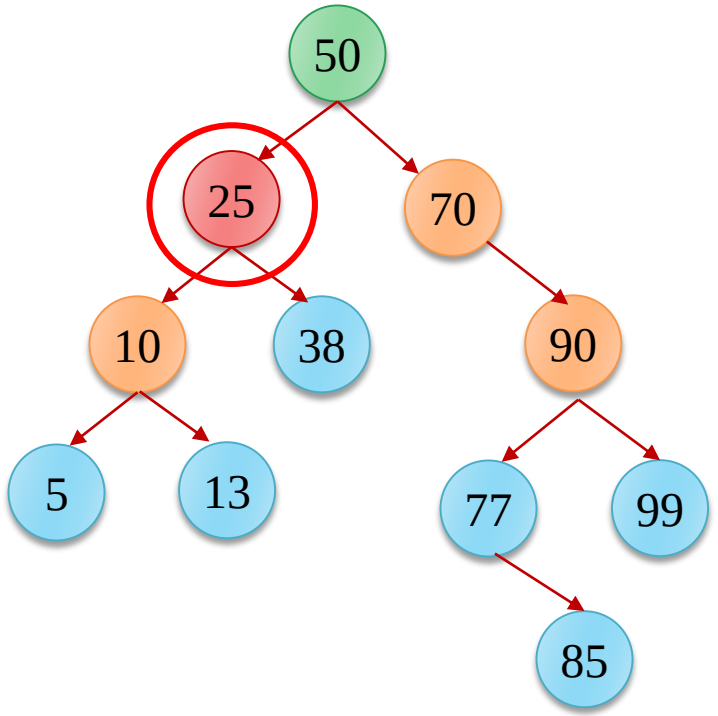
BST order property must be maintained.

What about 5? Order will be disturbed

What about 10? It has two children, where will 38 go?

38 is more appropriate choice?

What about 13? ok



Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 3: If Node has two child

Delete 30

Which node will become left child of 65

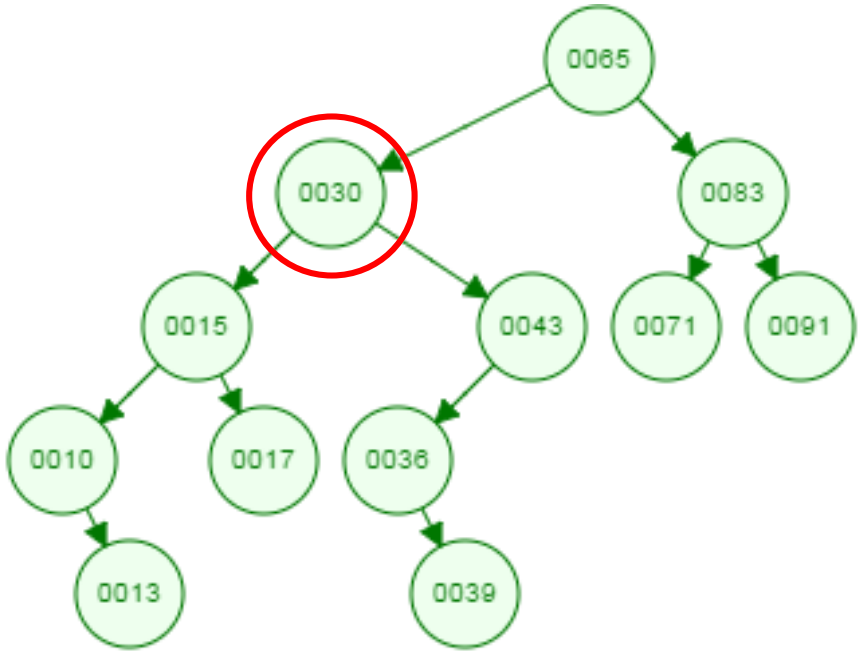
13? Order will be disturbed

15? Better choice is 17

No order disturbance, leaf node

43? Order will be disturbed

36? yes



Have you observed the pattern for choosing node?

Chosen node has single child – simple case

Or it is leaf node – simple case

- If we choose node from left sub-tree

  It will not have a right child and it is largest in left-sub tree

- If we choose from right sub-tree

  It will not have left-child and it is smallest in right sub-tree

See the next slide

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 3: If Node has two child

See the tree with few changes

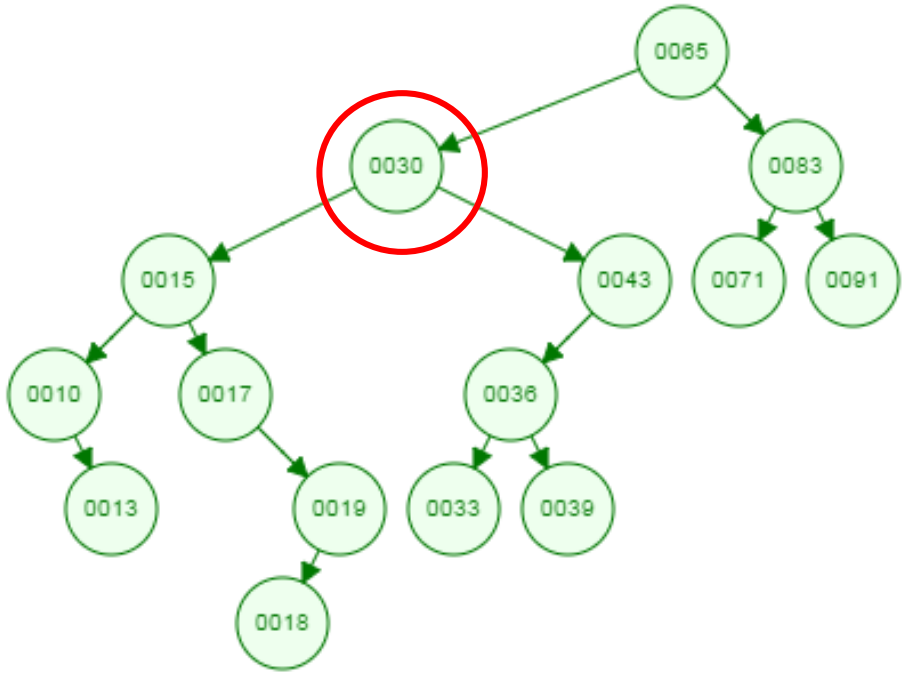
Now is it good to choose 17?

Order will be disturbed?

So now choose 19

- Same is the case with 36

Now 33 is choice



Actually if you do the in-order traversal of tree

10, 13, 15, 17, 18, 19, 30, 33, 36, 39, 43, 65, 71, 83, 91

Then your chosen node is either predecessor or successor of deleted node

- In-order **Successor** of a node is a node which comes after the node in in-order traversal.

In-order **predecessor** of a node is a node which comes before the node in in-order traversal.

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 3: If Node has two child

1. Either

   Find in-order successor of node

   Replace contents of node with successor node

   Delete successor

2. OR

   Find in-order predecessor of node

   Replace contents of node with predecessor node

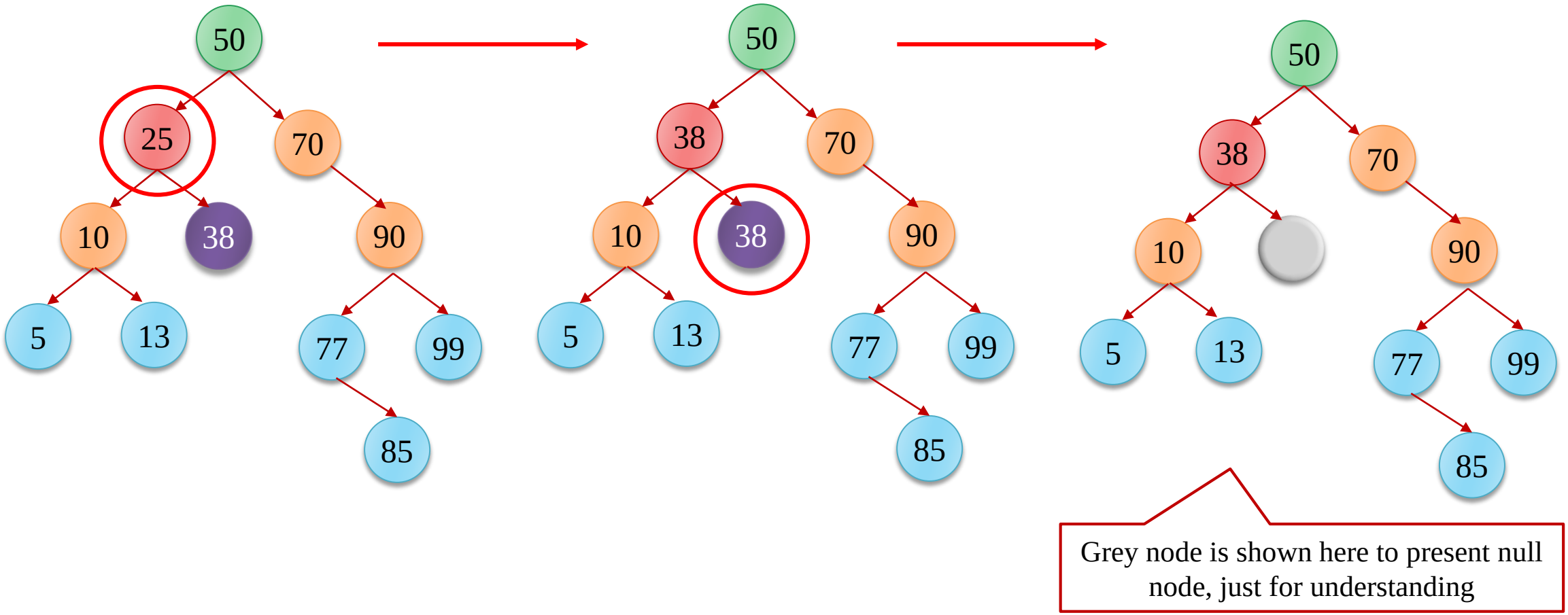
   Delete predecessor

- Next slides discuss the first approach.

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 3: If Node has two child

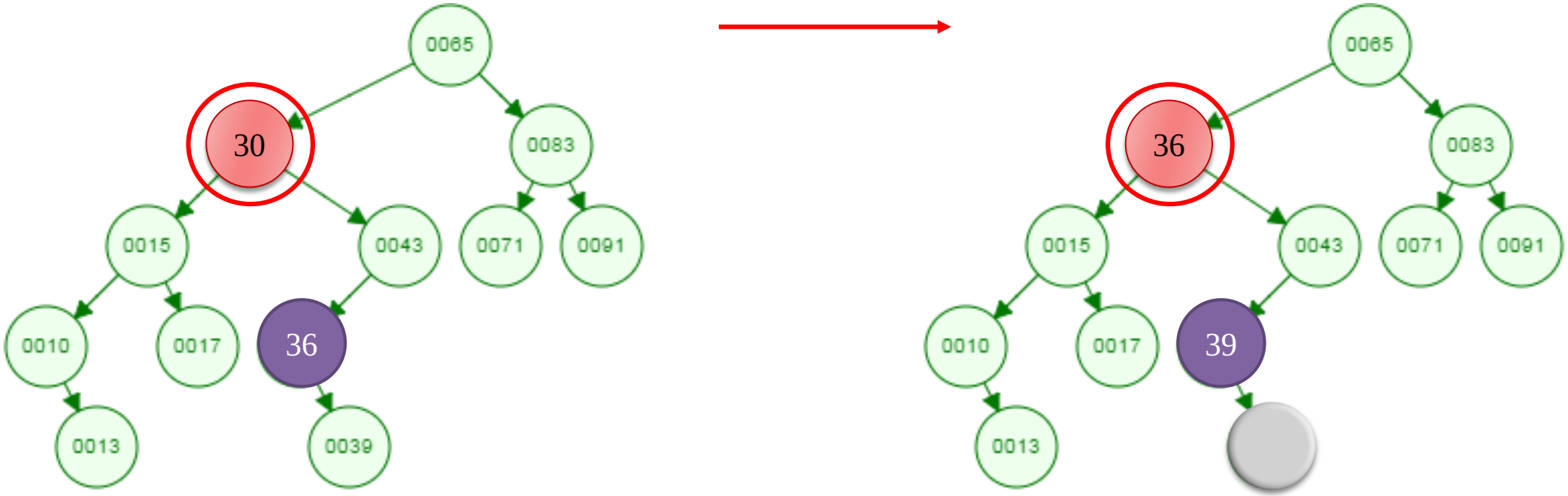


Grey node is shown here to present null node, just for understanding

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 3: If Node has two child



Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Find in-order successor of node?

INORDER_SUCCESSOR(node)

Set curr= node.right

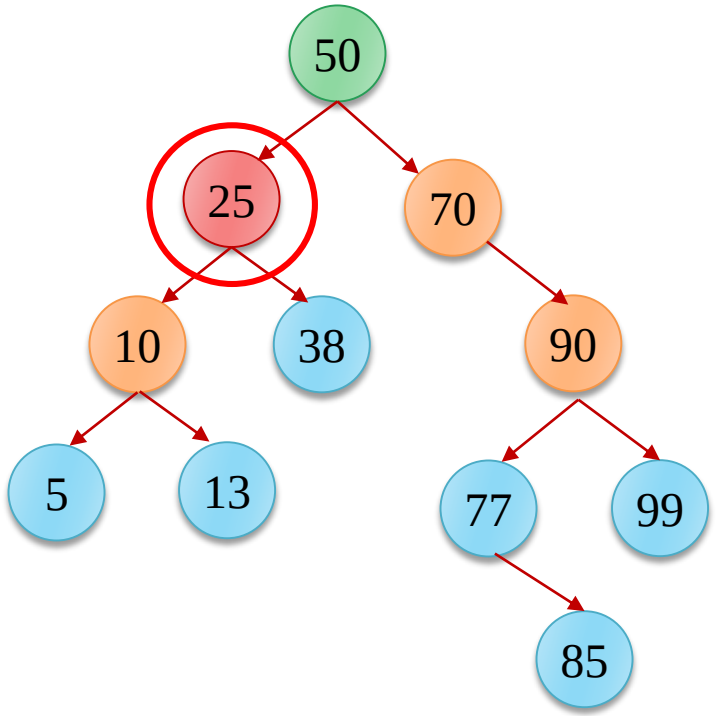
While(curr.left!=null)

curr=curr.left

End While

Return curr

For 25 it will return 38



BST_DELETE(node, int key)

Input: root node of tree, key to be deleted

Output: updated tree without node that contains key

Steps:

```
1.  If(node!=null)
2.      If key ==node.key
3.          If hasLeft(node) and hasRight(node)   //case 3
4.              successor= INORDER_SUCCESSOR(node)
5.              copy(node,successor)
6.              node.right= BST_DELETE(node.right, successor.key )
7.          Else //case1, case 2, you can break it in 2 if's
8.              node=getChildNode(node);
9.          End If
9.      Else If key < node.key
10.         node.left =BST_DELETE(node.left,key)
11.     Else        // key > node.key
12.         node.right =BST_DELETE(node.right,key)
13.     End If
14.     return node  // updated node
15. End If
16. return node // null
```

getChildNode(node)

1. If hasLeft(node)
2. return node.left
3. Else If hasRight(node)
4. return node.right
5. Else
6. Return NULL// leaf node case
7. End if

copy(node,successor)

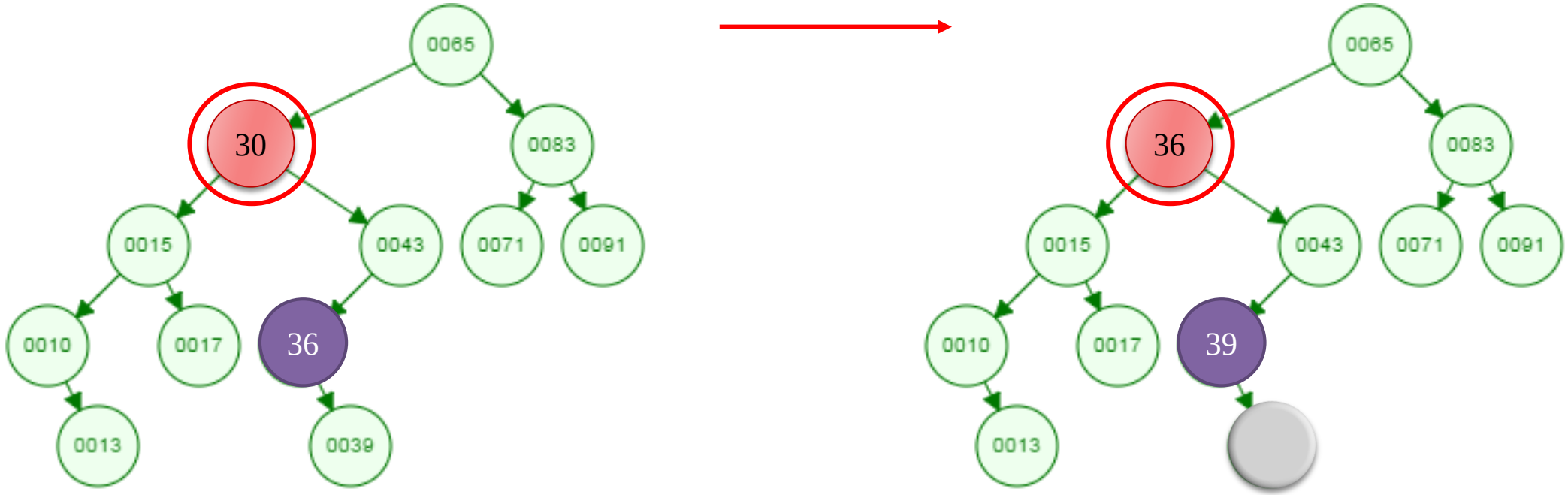
1. node.key=successor.key
2. Copy other data attributes if present but not child links

Edited with the trial version of Foxit Advanced PDF Editor
To remove this notice, visit: www.foxitsoftware.com/shopping

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

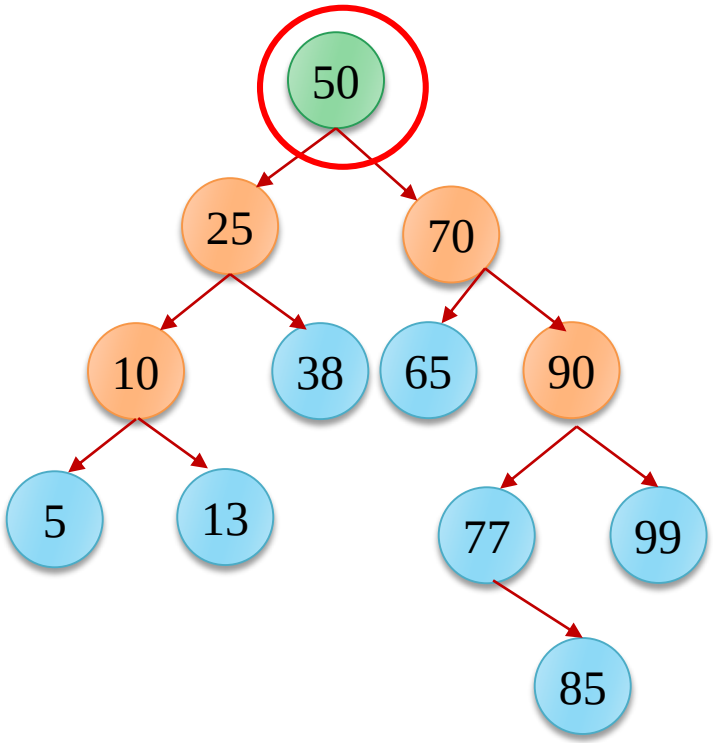
Case 3: If Node has two child

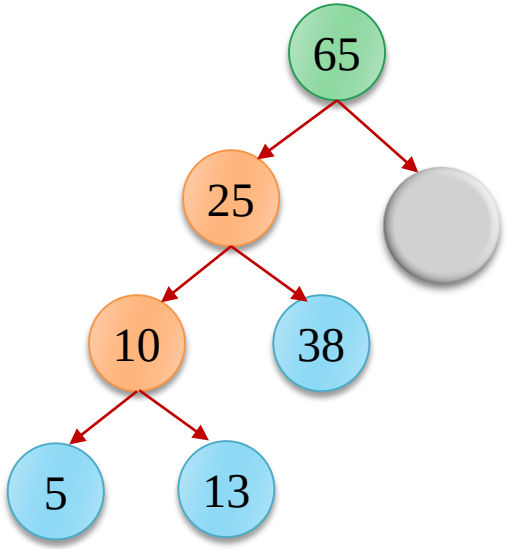


Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Explaination

node.right= **BST_DELETE**(successor, successor.key)





So rather than jumping to successor node, start from most immediate right node, and systematically delete successor node

Reason: when function is called with successor as root, it store child of successor to right of current node, which in this case is null, and sub-tree starting at 70 is removed

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

Deletion

Case 3: If Node has two child

Even though it does not affect BST property if you choose successor or predecessor, but it is good to choose randomly between them to maintain tree balance

Otherwise.

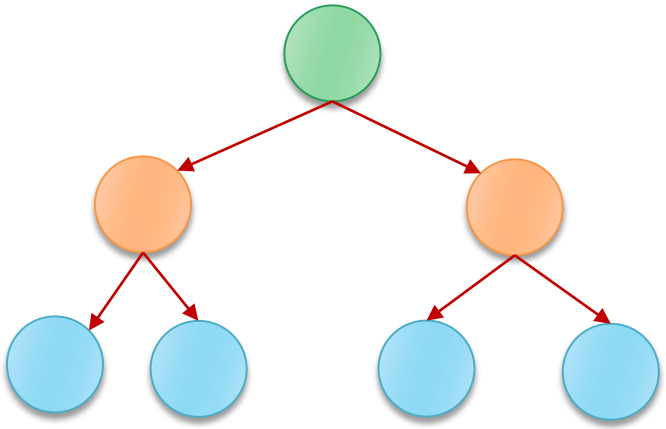
Tree will become skewed. One side will become extremely short than other

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

BST Time Complexity

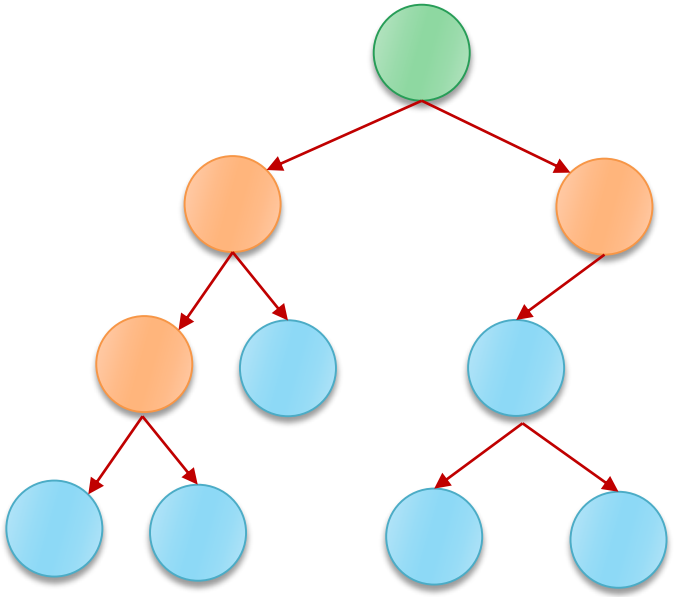
Depends upon height

Big-Oh➔h



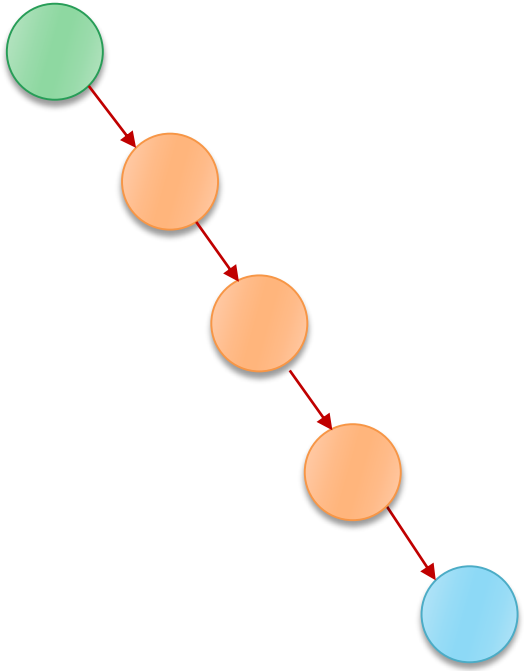
Best

h=log n



Average

h=almost log n



Worst

h=n

Edited with the trial version of
Foxit Advanced PDF Editor
To remove this notice, visit:
www.foxitsoftware.com/shopping

HomeWork

For every algorithm, root node is given as input along with any other input (if mentioned).

- To find and return parent of given node?
- To find and return depth/level of given node?
- To find and return height of tree
- To find if a tree is BST or not?
- To find if two nodes are at same level of tree or not?
- To find and return total number of nodes?