Sunbeam Institute of Information Technology

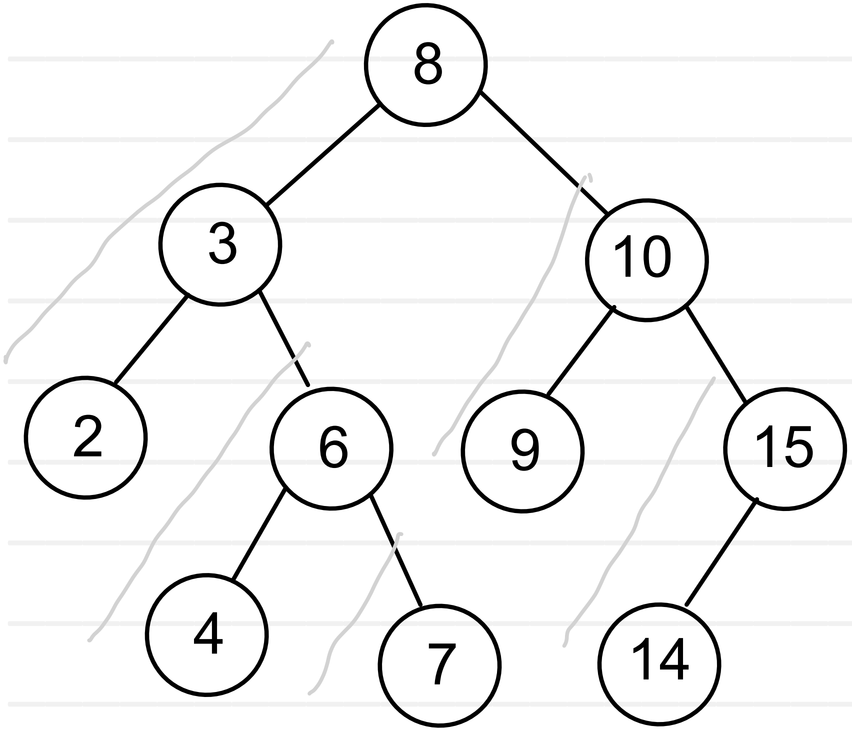
Pune and Karad

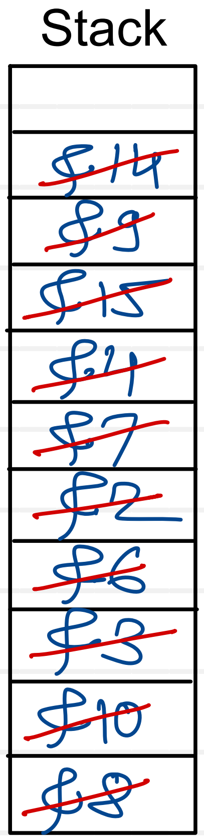
Module – Data Structures and Algorithms

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Binary Search Tree - DFS Traversal



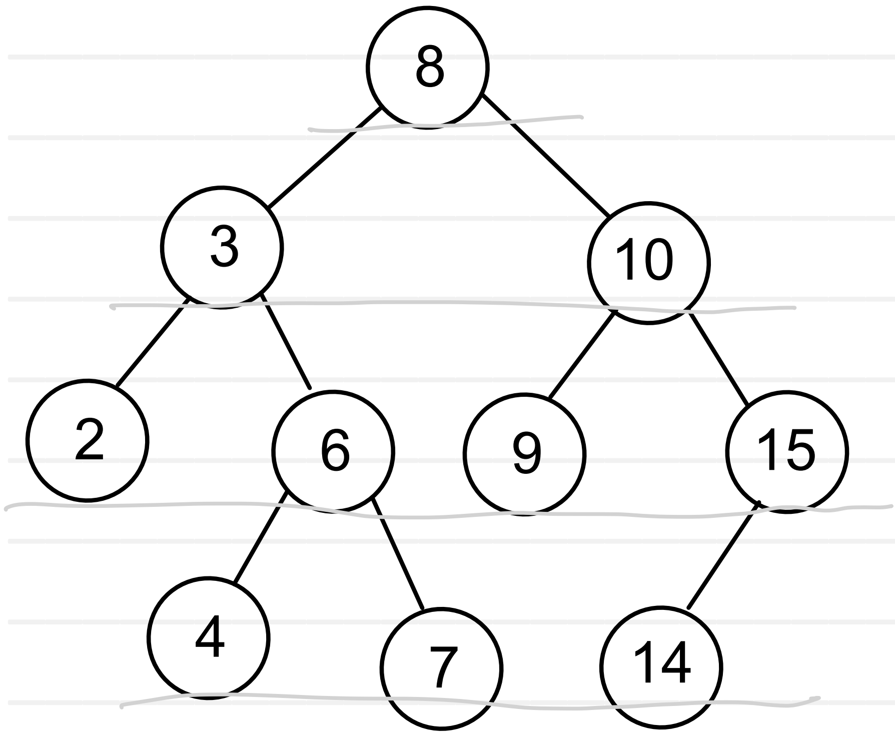


(Depth First Search)

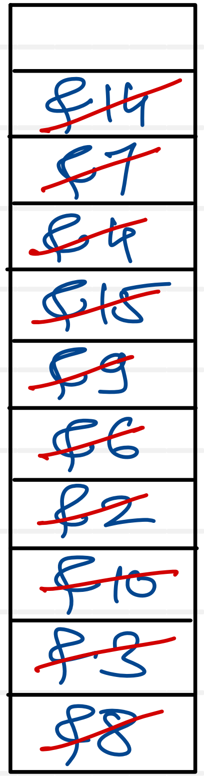
1. Push root node on stack
2. Pop one node from stack
3. Visit (print) popped node
4. If right exists, push it on stack
5. If left exists, push it on stack
6. While stack is not empty, repeat step 2 to 5

Traversal: 8, 3, 2, 6, 4, 7, 10, 9, 15, 14

Binary Search Tree - BFS Traversal



Queue

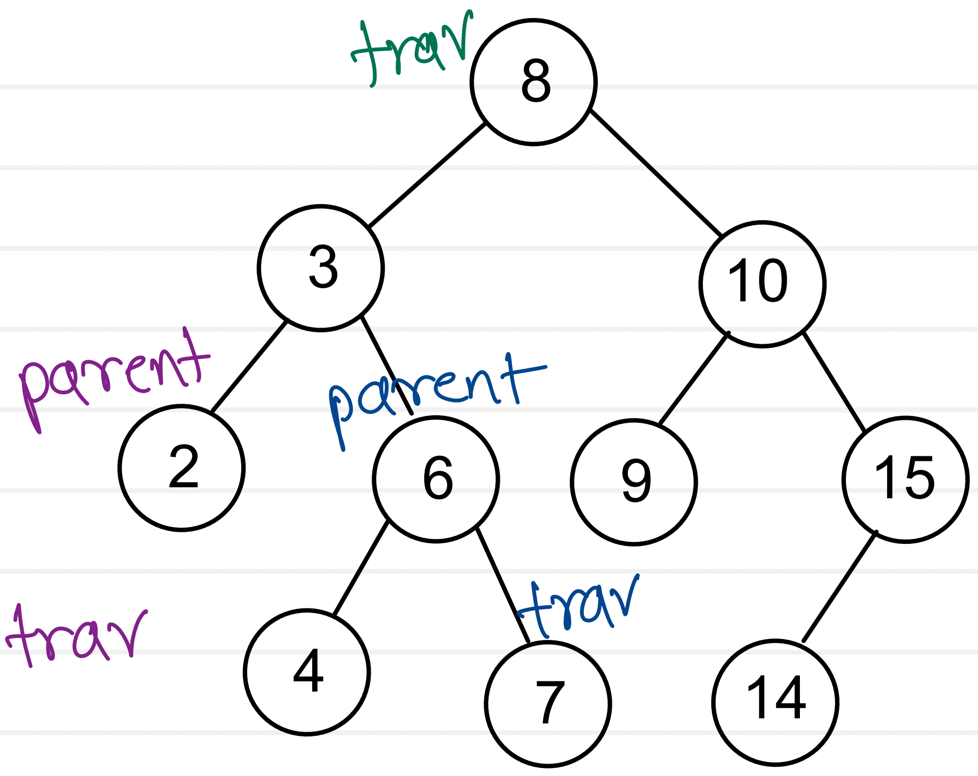


(Bredth First Search)

1. Push root node on queue
2. Pop one node from queue
3. Visit (print) popped node
4. If left exists, push it on queue
5. If right exists, push it on queue
6. While queue is not empty, repeat step 2 to 5

Traversal : 8 , 3 , 10 , 2 , 6 , 9 , 15 , 4 , 7 , 14

Binary Search Tree - Binary Search with Parent



Key=1

trav	parent
&8	null
&3	&8
&2	&3
null	&2

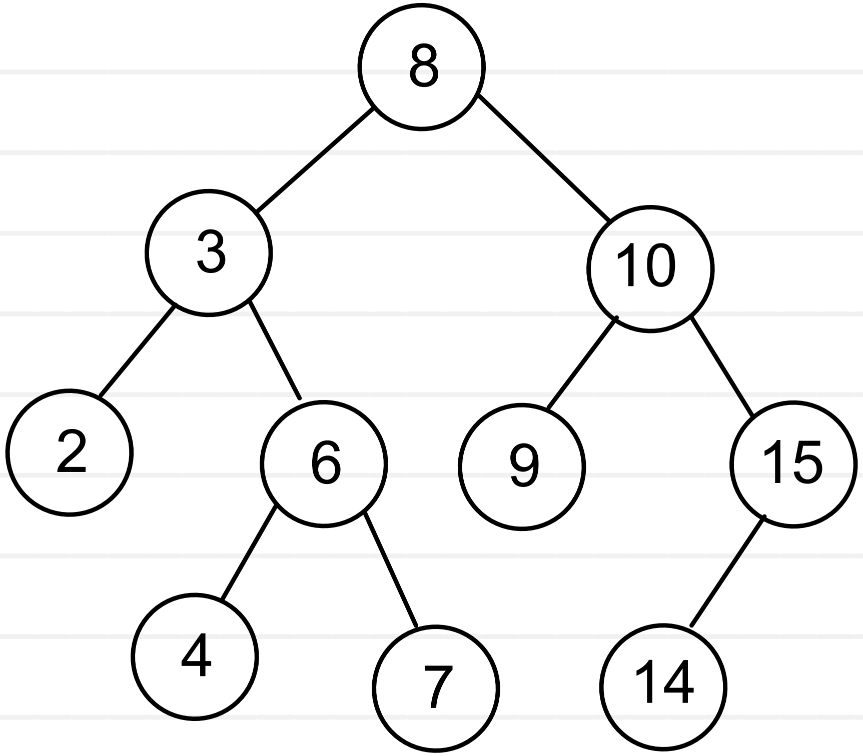
key=7

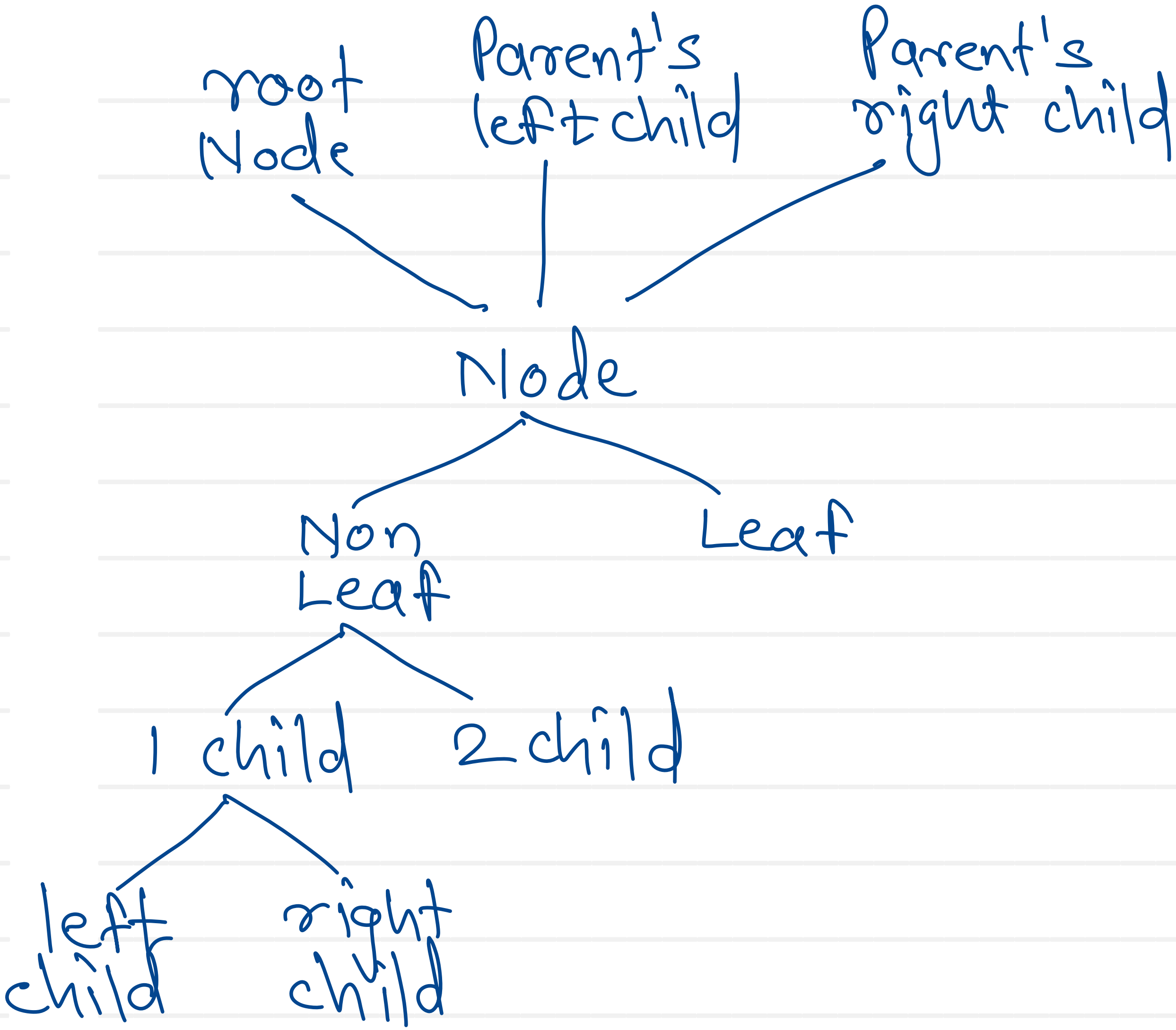
trav	parent
&8	null
&3	&8
&6	&3
&7	&6

Key=8

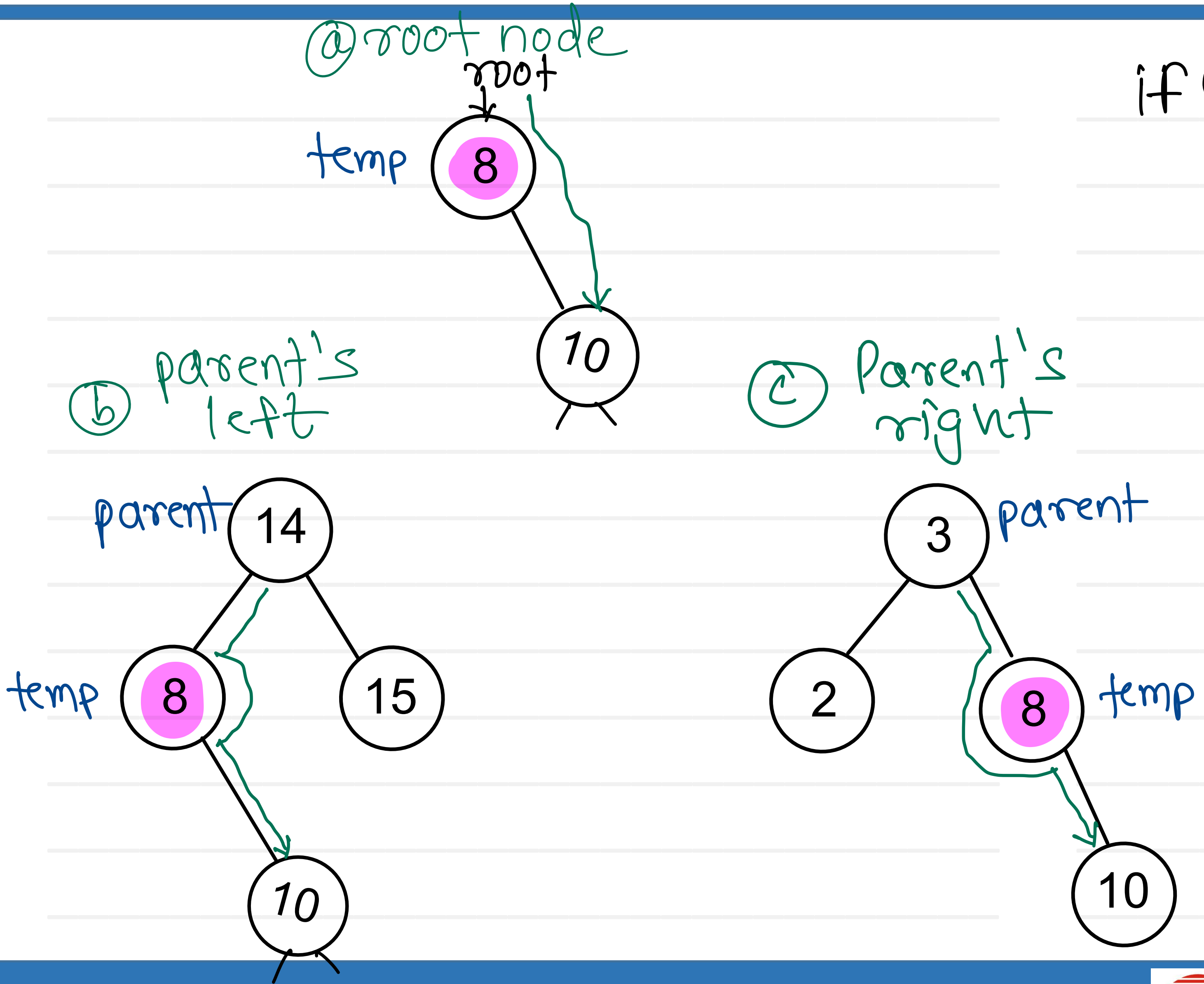
trav	parent
&8	null

Binary Search Tree - Delete Node





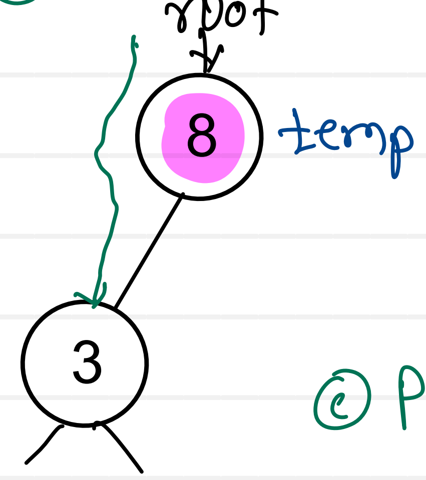
BST - Delete Single child node (Right child)



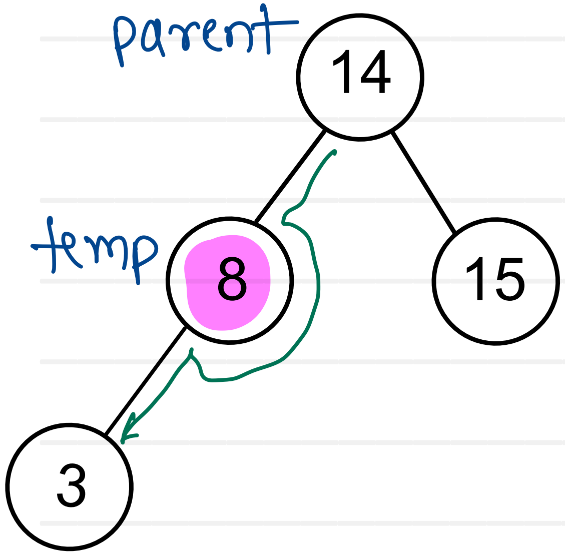
```
if ( temp.left == null ) {
        if(temp == root)
            root = temp.right
        else if(temp == parent.left)
            parent.left = temp.right;
        else if(temp == parent.right)
            parent.right = temp.right;
}
```

BST- Delete Single child node (Left child)

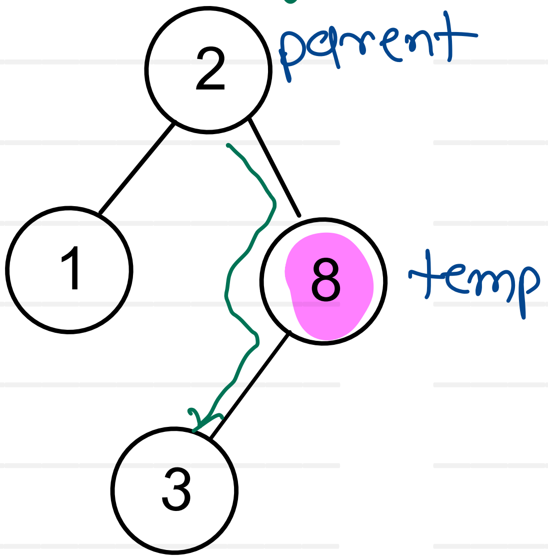
ⓐ root node



ⓑ parent's left



ⓒ parent's right



```
if(temp.right == null) {
    if(temp == root)
        root = temp.left;
    else if(temp == parent.left)
        parent.left = temp.left;
    else if(temp == parent.right)
        parent.right = temp.left;
}
```

SUNBEAM

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com