DATA STRUCTURES AND ITS APPLICATIONS

Splay Tree

Kusuma K V

Department of Computer Science & Engineering

DATA STRUCTURES AND ITS APPLICATIONS

Splay Tree

Kusuma K V

Department of Computer Science & Engineering

Splay Tree

Splaying

- There are three types of splay steps, each of which has two symmetric variants: left- and right-handed.
- Zig step: this step is done when p is the root.
- Zig - Zig step: this step is done when p is not the root and x and p are either both right children or are both left children.
- Zig - Zag step: this step is done when p is not the root and x is a right child and p is a left child or vice versa (x is left, p is right).

Deletion

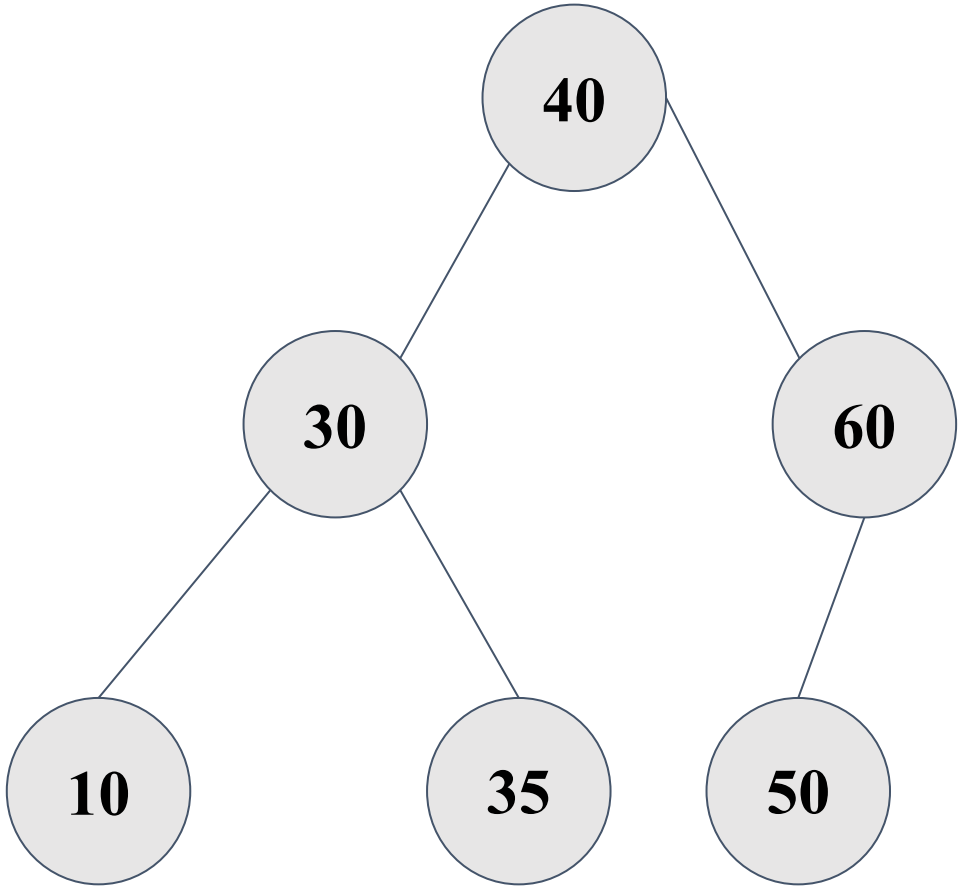
To delete a node *x*, use the same method as with a binary search tree:
If *x* has two children:

- Swap its value with that of either the rightmost node of its left sub tree (its in-order predecessor) or the leftmost node of its right subtree (its in-order successor).
- Remove that node instead.

In this way, deletion is reduced to the problem of removing a node with 0 or 1 children.
Unlike a binary search tree, in a splay tree after deletion, we splay the parent of the removed node to the top of the tree.

Splay Tree Deletion

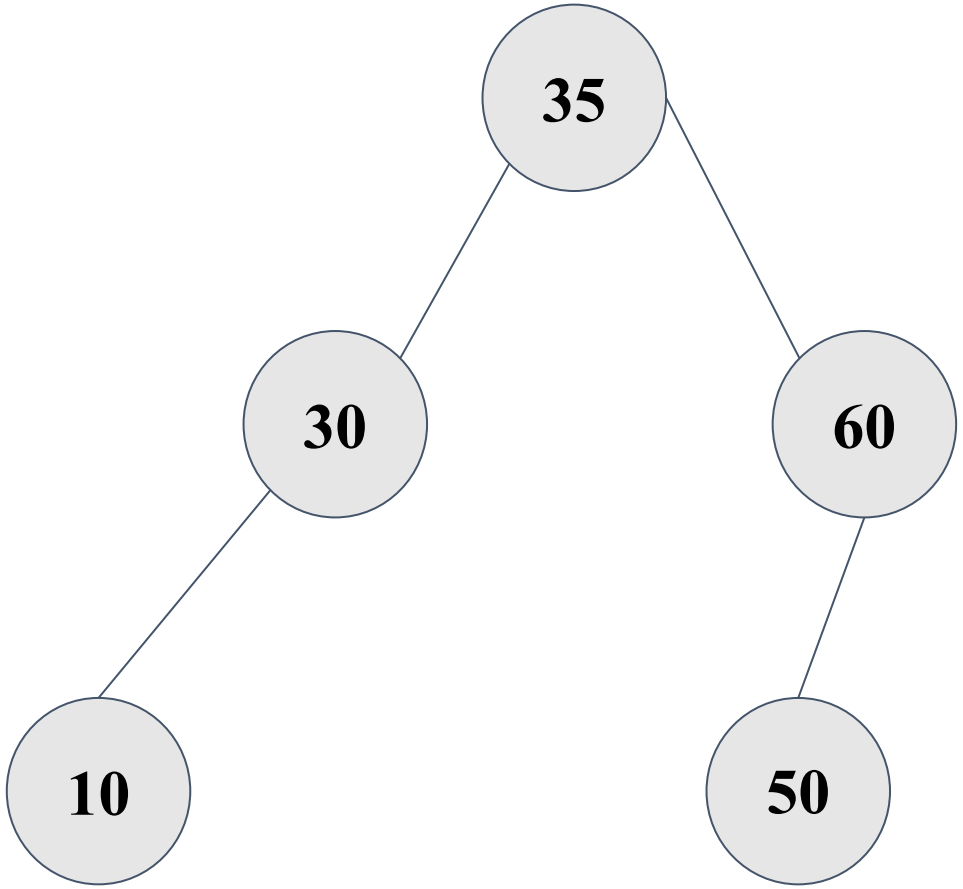
Delete 40, 10, 35, 70



Initial Tree

Splay Tree Deletion

Delete 40, 10, 35, 70

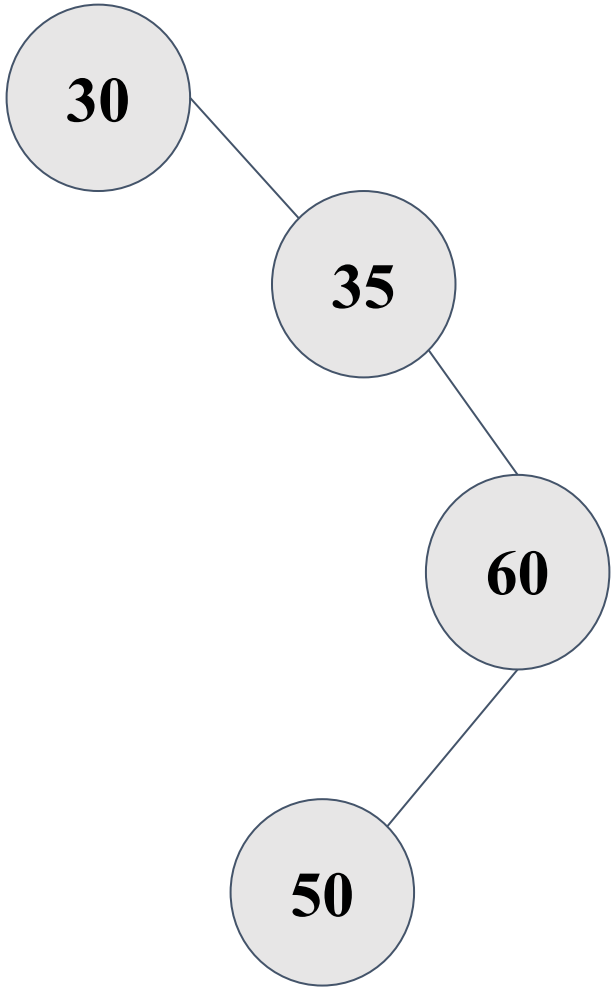


Delete 40

- Splay node 40 to root (already root).
- Remove root node 40.
- Split left subtree (root 30) and right subtree (root 60).
- Splay max node in left subtree (35) to root.
- Attach right subtree (60) as right child of 35.

Splay Tree Deletion

Delete 40, 10, 35, 70

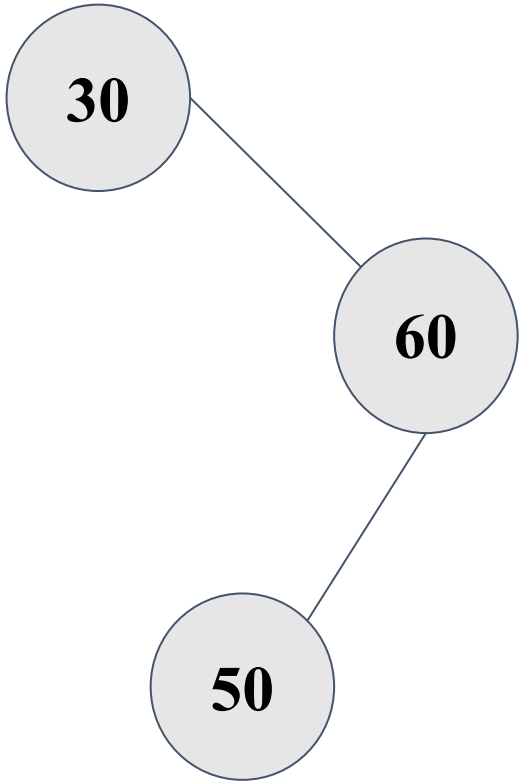


Delete 10

- Splay 10 to root.
- Remove root (10).
- Left subtree empty, so root becomes right subtree (30 subtree).

Splay Tree Deletion

Delete 40, 10, 35, 70

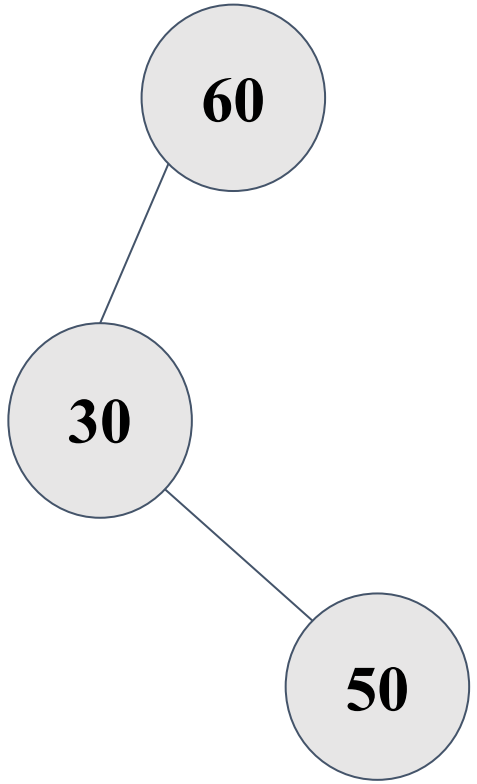


Delete 35

- Splay 35 to root.
- Remove root (35).
- Split into left subtree (30) and right subtree (60).
- Splay max in left subtree (30) to root.
- Attach right subtree (60).

Splay Tree Deletion

Delete 40, 10, 35, 70

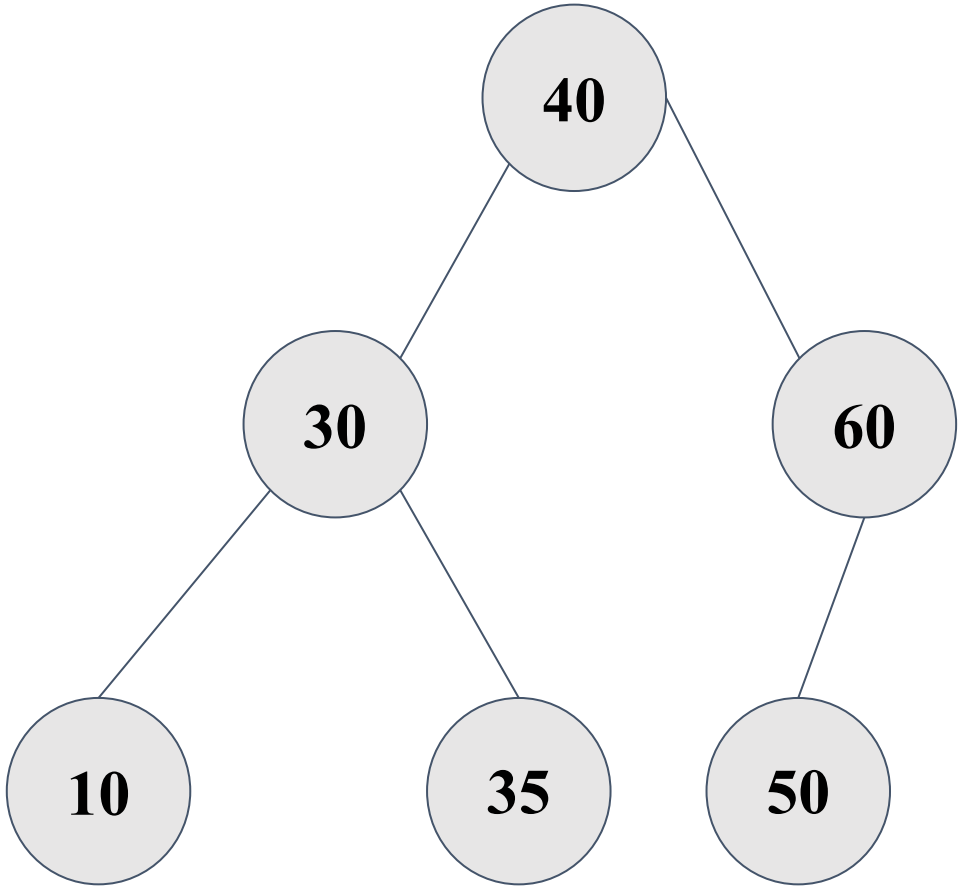


Delete 70

- Search and splay 70.
- 70 not found, splay last accessed node (likely 60) to root.
- Since 70 does not exist, tree remains unchanged.

Splay Tree Search

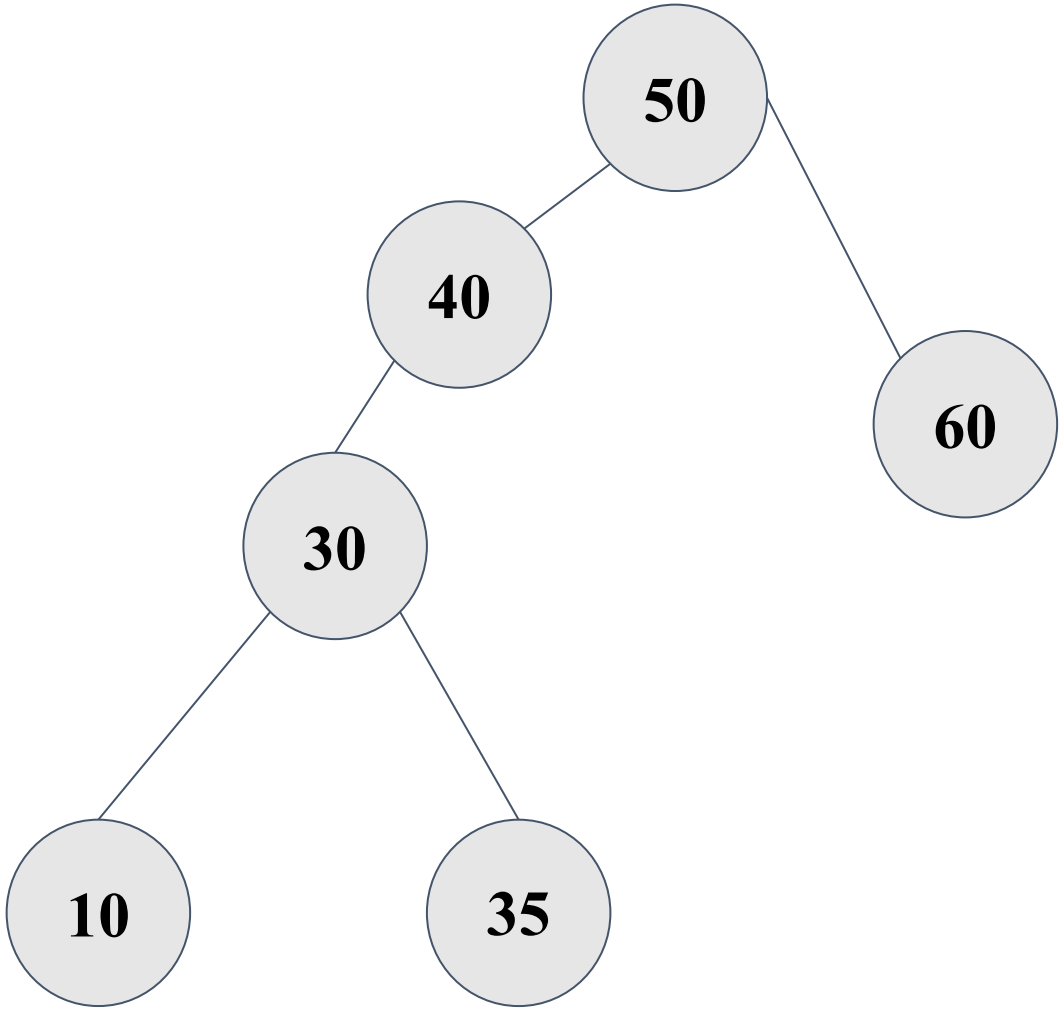
Search 50, 20



Initial Tree

Splay Tree Search

Search 50, 20

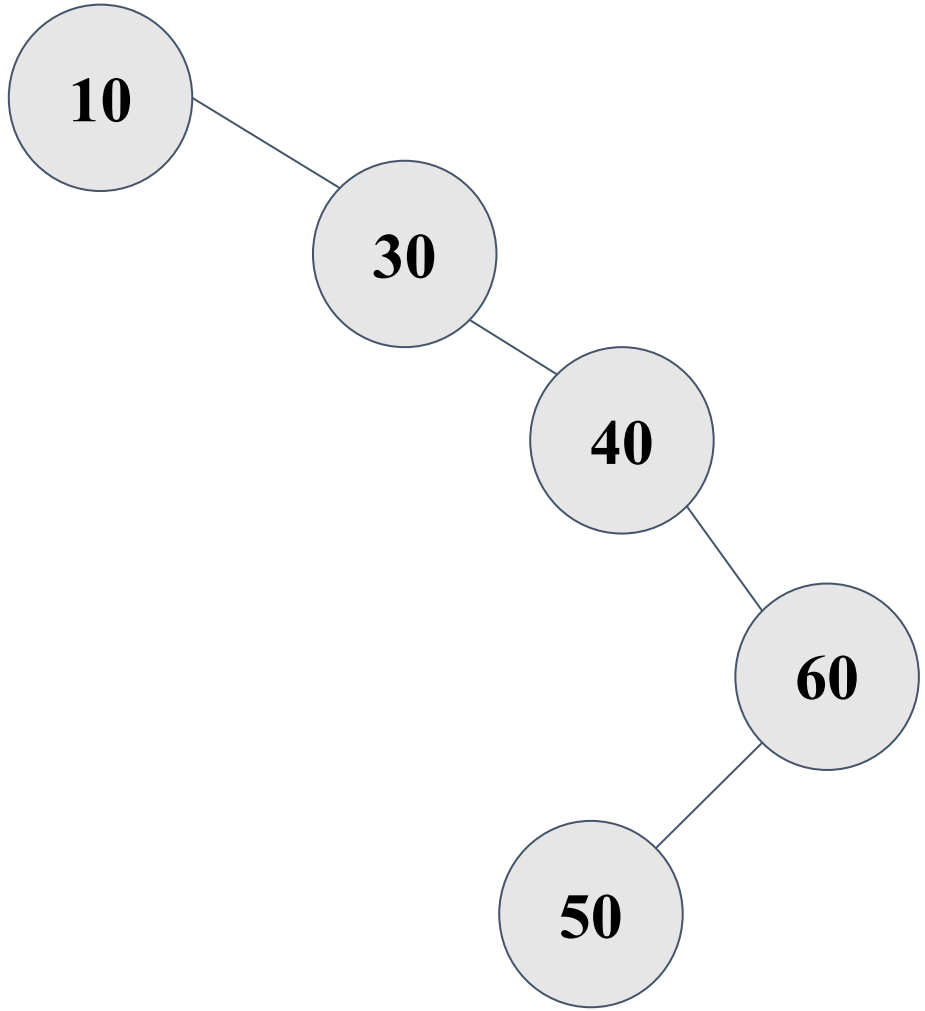


Search 50 and Splay it to Root

- Search for node 50.
- Node 50 found as left child of 60.
- Splay 50 to root by rotations:
  1. Rotate 50 up with 60.
  2. Rotate 50 up with 40.

Splay Tree Search

Search 50, 20



Search 20 (not found)

- Traverse: 40 -> 30 -> 10 -> null (20 not found)
- Splay last accessed node 10 to root

References

- https://en.wikipedia.org/wiki/Splay_tree
- "Data Structures and Program Design in C", Robert Kruse, Bruce Leung, C.L Tondo, Shashi Mogalla, Pearson, 2nd Edition, 2019.

THANK YOU

Kusuma K V

Department of Computer Science & Engineering