Unit 9 – BST, Treap

CS 201 - Data Structures II
Spring 2023
Habib University

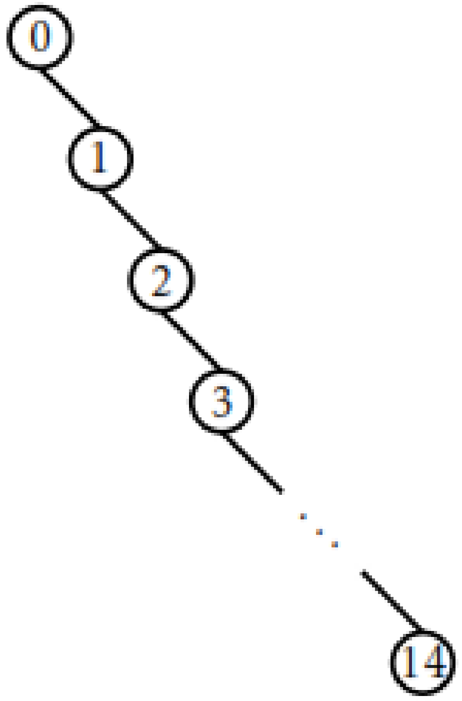
Syeda Saleha Raza

Recap

- Tree
- Binary Tree
  - Terminologies
  - Traversal (Inorder, pre-order, post-order)
- Binary Search Tree
  - Binary Search property
  - Find/Add/Remove node

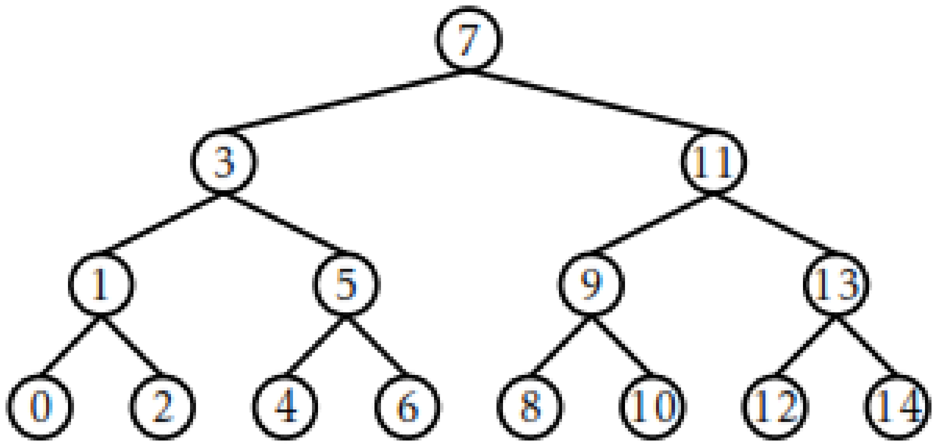
BST

$$\langle 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 \rangle \ .$$



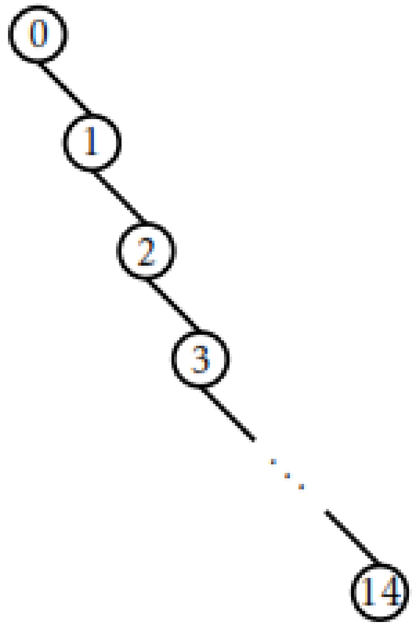
BST

$\langle 7, 3, 11, 1, 5, 9, 13, 0, 2, 4, 6, 8, 10, 12, 14 \rangle$.

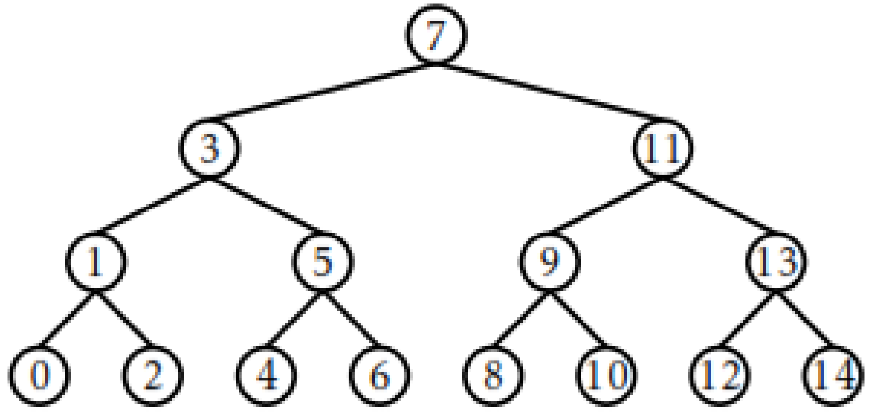


Binary Search Tree on same data

$\langle 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 \rangle$.

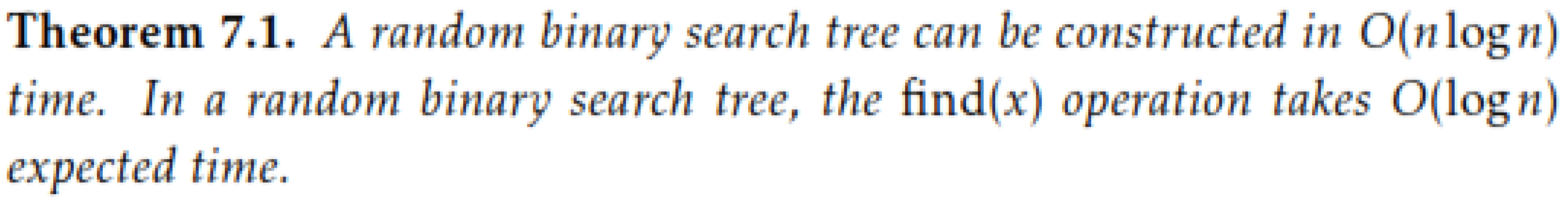


$\langle 7, 3, 11, 1, 5, 9, 13, 0, 2, 4, 6, 8, 10, 12, 14 \rangle$.



- In fact, there are 21*964; 800 addition sequences that generate the tree on* the right and only one that generates the tree on the left.

Randomness in BST



Theorem 7.1. *A random binary search tree can be constructed in* $O(n \log n)$ *time. In a random binary search tree, the* find(x) *operation takes* $O(\log n)$ *expected time.*

Treap
(A blend of Tree + Heap)

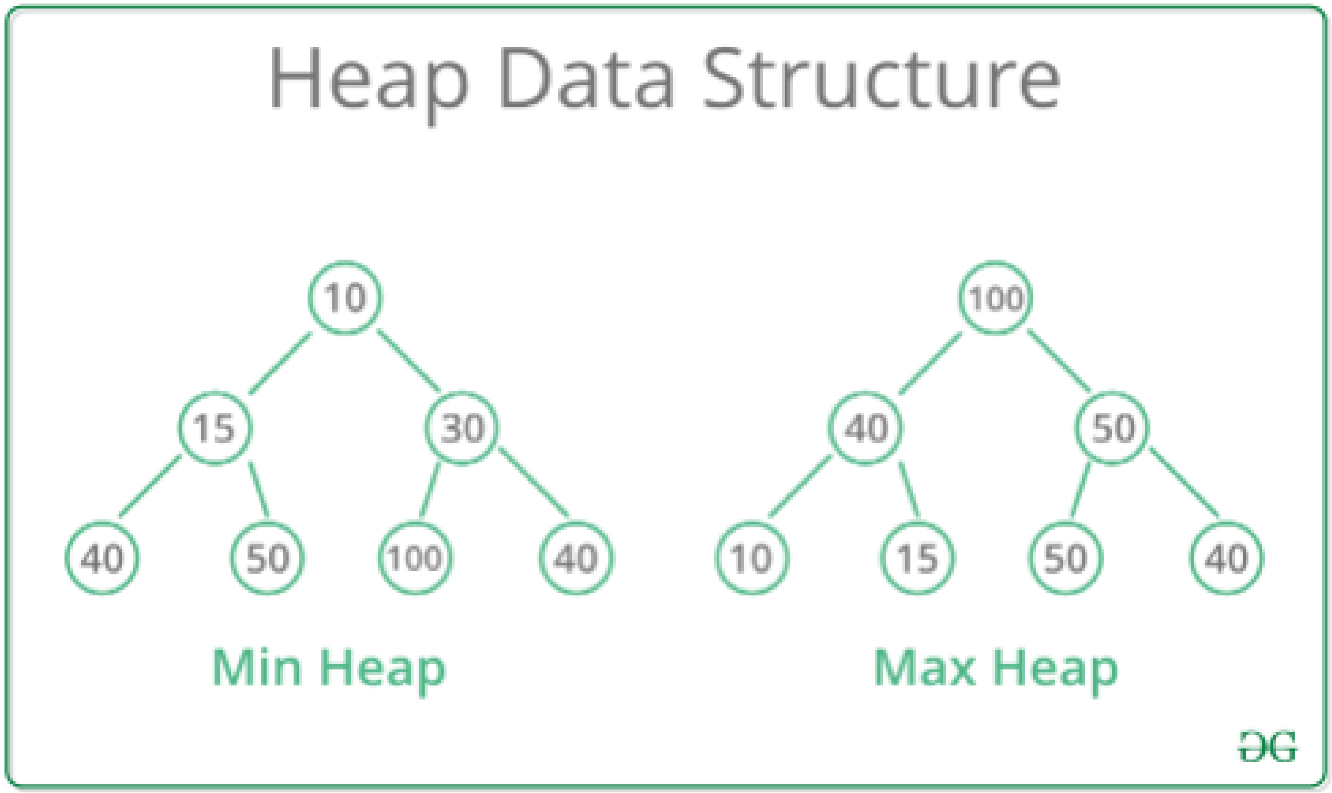
Treap - Randomized Binary Search Tree

Every node of Treap maintains two values:

1. **Key** Follows standard BST ordering (left is smaller and right is greater)
2. **Priority** Randomly assigned value that follows Max-Heap property.

Treap (A Randomized Binary Search Tree) - GeeksforGeeks

Heap Property



Heap Property

- Each node has a priority smaller than that of its two children. An example is shown in Figure7.5.
- The heap and binary search tree conditions together ensure that, once

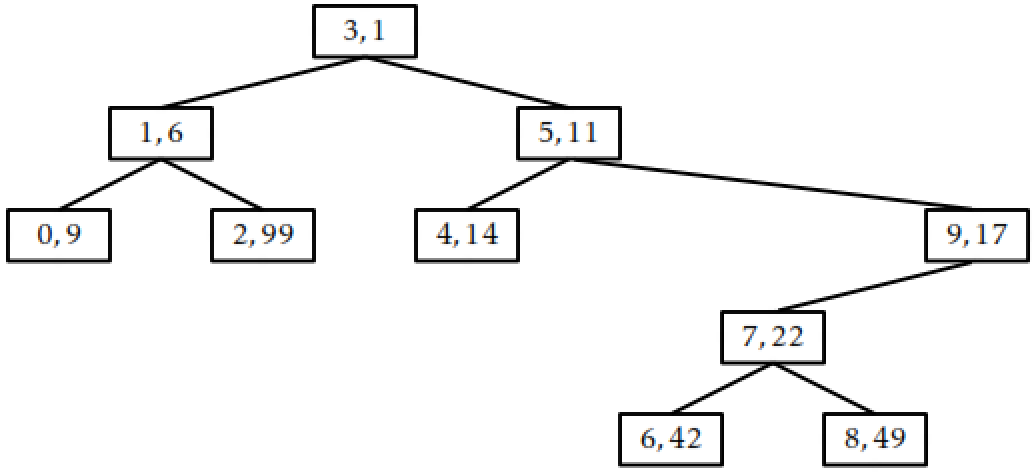


Figure 7.5: An example of a Treap containing the integers 0,...,9. Each node, u, is illustrated as a box containing $u.x, u.p$.

Treap Properties

- The **heap property** tells us that the node with minimum priority has to be the root, r, of the Treap.
- The **binary search tree property** tells us that all nodes with keys smaller than the current key are stored in the left subtree rooted and all nodes with keys larger than the current key are stored in the right subtree.
- The important point about the priority values in a Treap is that they are **unique** and **assigned at random**.
- The heap and binary search tree conditions together to ensure that **the shape of the Treap is completely determined**.

Treap

- We can think of a Treap as a:
  - Tree that obeys heap and binary search property.
  - Binary Search Tree whose nodes were added in increasing order of priority.

$$\langle(3,1),(1,6),(0,9),(5,11),(4,14),(9,17),(7,22),(6,42),(8,49),(2,99)\rangle$$

$$\langle 3,1,0,5,9,4,7,6,8,2\rangle$$

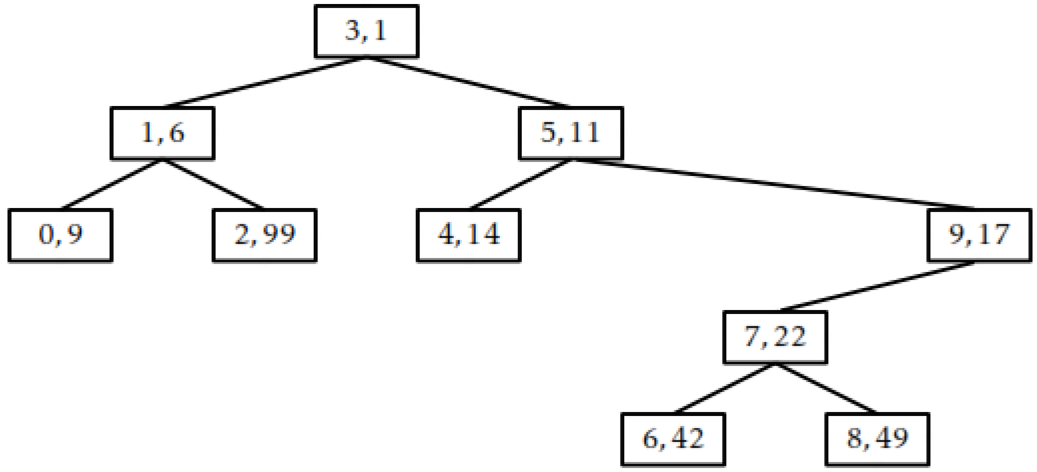
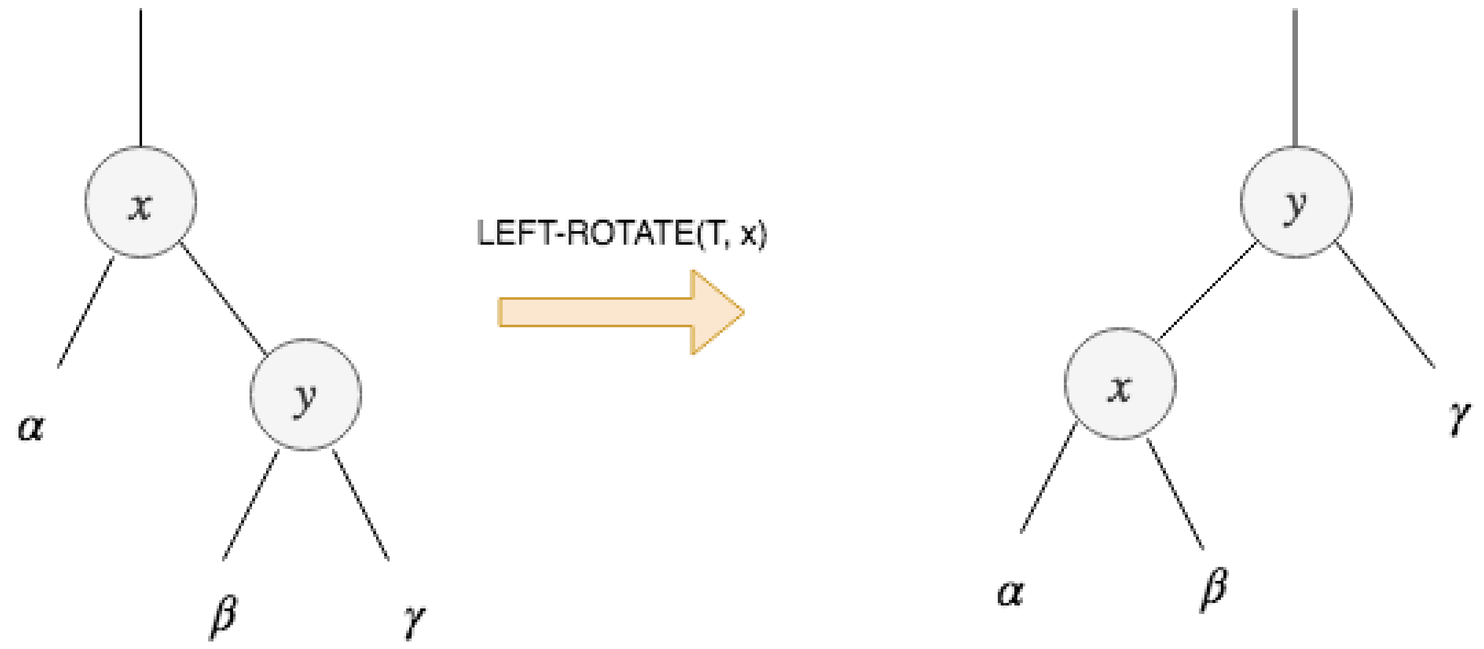


Figure 7.5: An example of a Treap containing the integers 0,...,9. Each node, u,

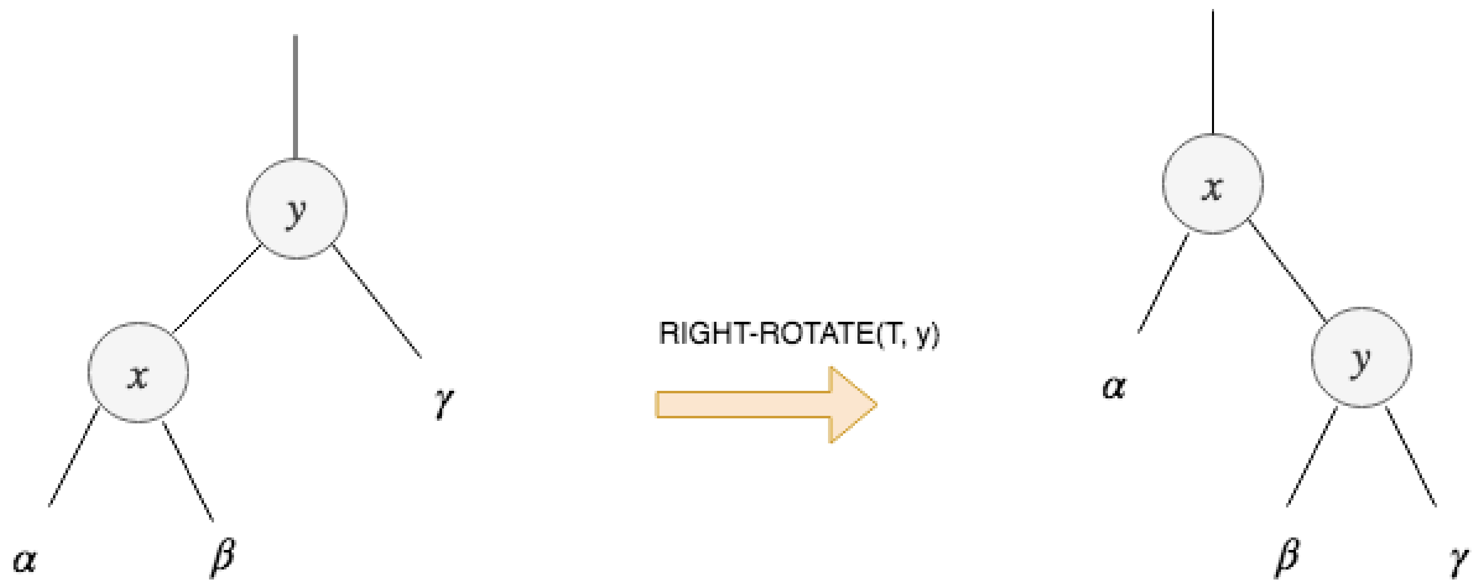
Tree Rotation

- Tree rotation is a transformation technique which can be used to change the structure of the binary search tree *without changing the order of the elements*.
- Rotation supports in balancing the by reducing the depth of a node by one and increases the depth of its parent by one.

Left Rotation



Right Rotation



Rotation in Tree

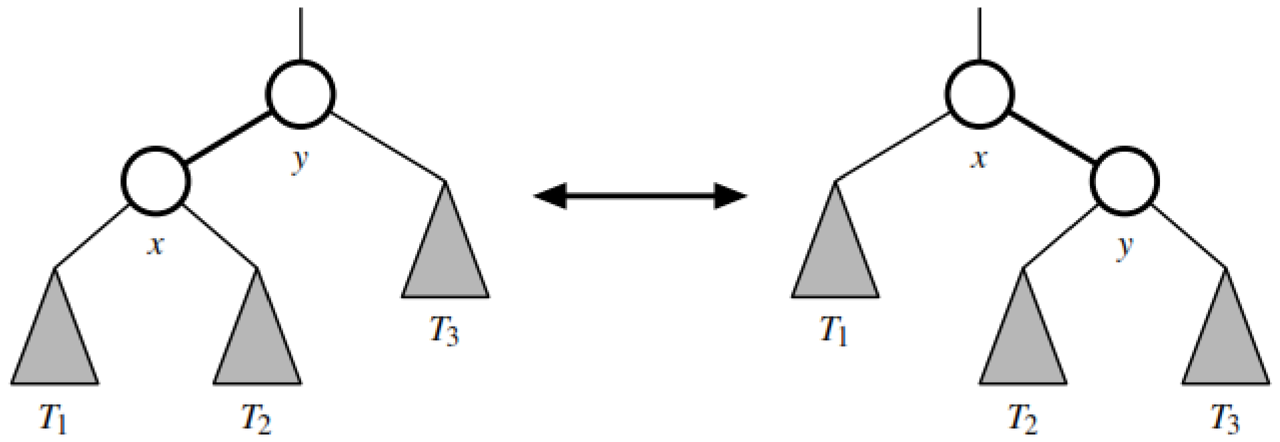
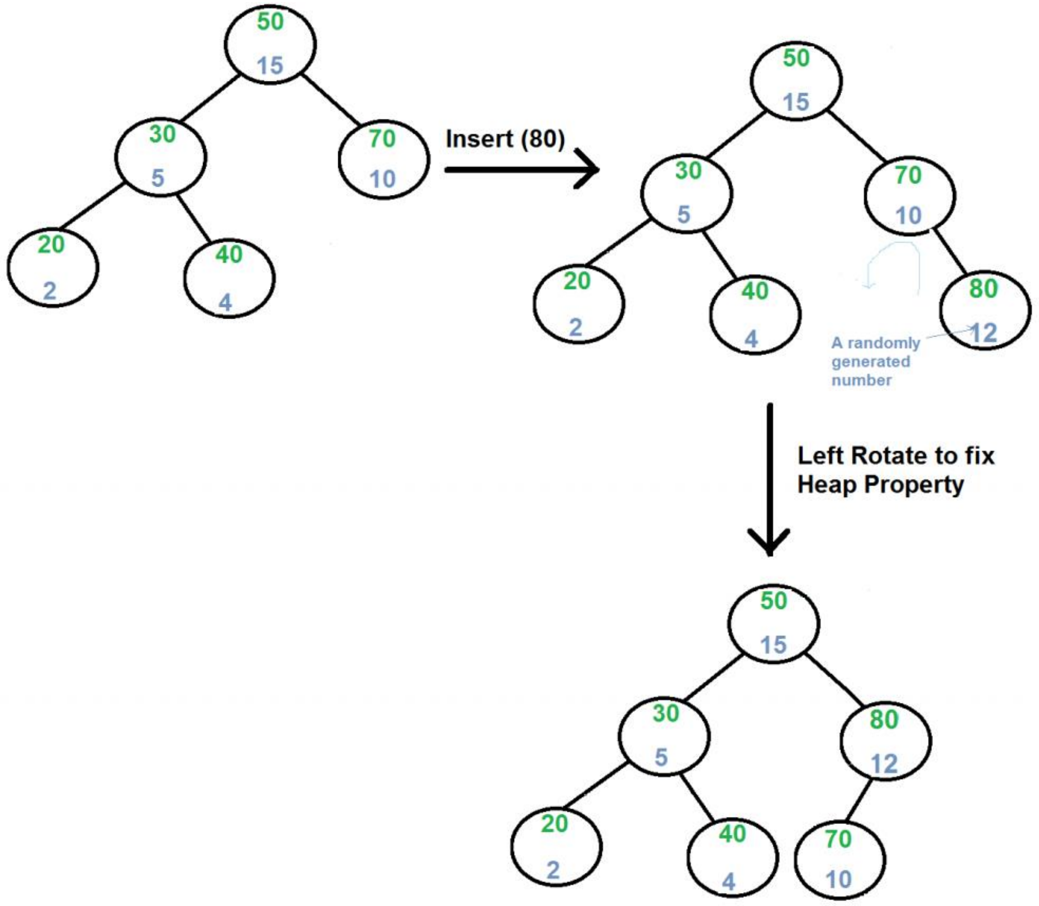


Figure 11.8: A rotation operation in a binary search tree. A rotation can be performed to transform the left formation into the right, or the right formation into the left. Note that all keys in subtree T_1 have keys less than that of position x, all keys in subtree T_2 have keys that are between those of positions x and y, and all keys in subtree T_3 have keys that are greater than that of position y.

Insertion in Treap

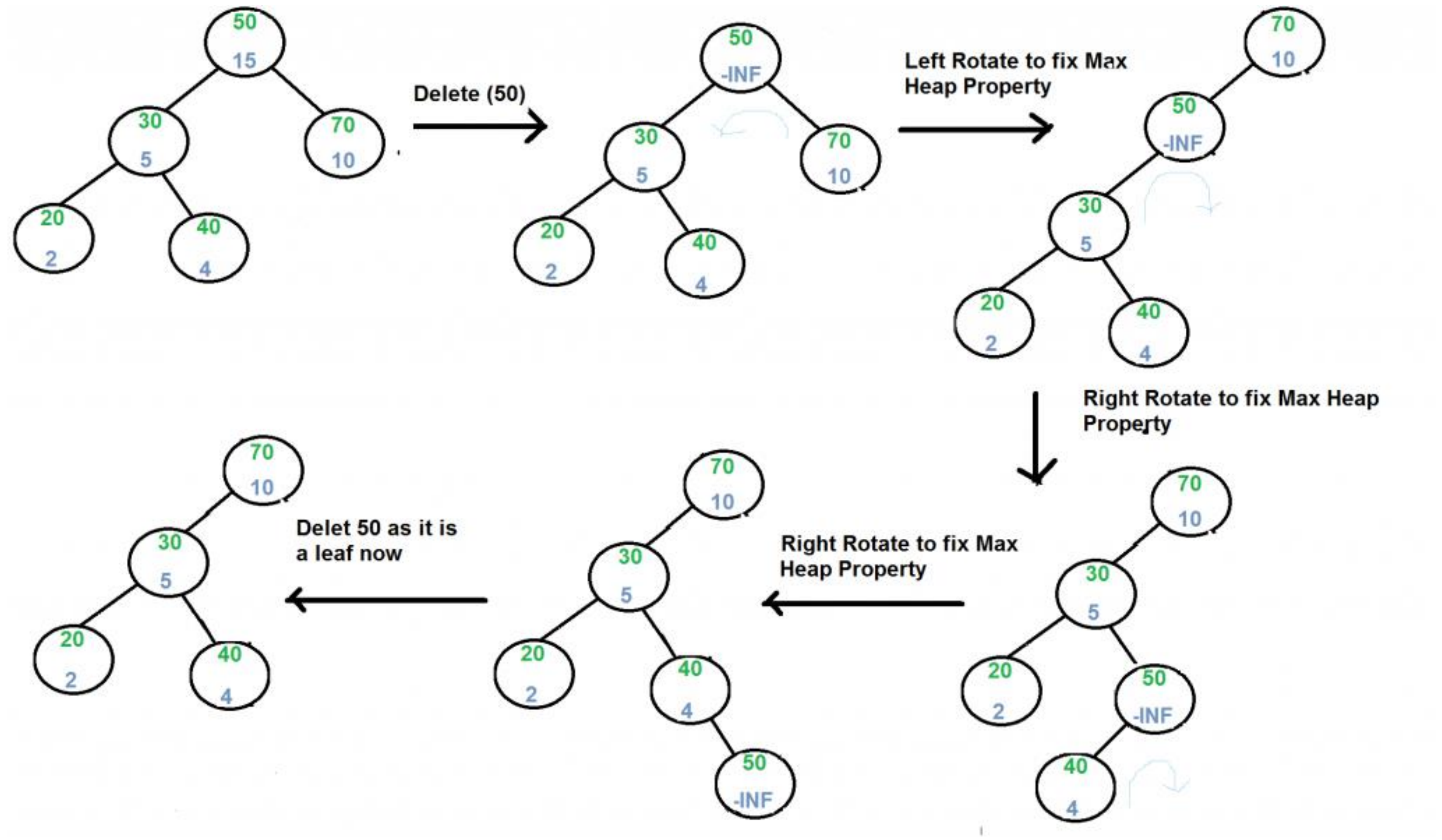


[Treap (A Randomized Binary Search Tree) - GeeksforGeeks](https://www.geeksforgeeks.org/treap-a-randomized-binary-search-tree/)

Deletion in Treap

- If node to be deleted is a leaf, delete it.
- Else replace node's priority with minus infinite (-INF), and do appropriate rotations to bring the node down to a leaf.

Deletion in Treap



50 15
30 5
70 10
20 2
40 4
Delete (50)
50 -INF
Left Rotate to fix Max Heap Property
Right Rotate to fix Max Heap Property
Right Rotate to fix Max Heap Property
Delet 50 as it is a leaf now

[Treap (A Randomized Binary Search Tree) - GeeksforGeeks]

Treap vs Skiplist

- Both can be used for SSet interface
- Find(), add(), remove()

Resources

- Open Data Structures (pseudocode edition), by Pat Morin. Available online at http://opendatastructures.org
- Data Structures and Algorithms in Python, by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser. 2013. (1st. ed.). Wiley Publishing
- https://www.geeksforgeeks.org/treap-a-randomized-binary-search-tree/