BITS Pilani
Hyderabad Campus

Data Structures and Algorithms Design

Febin.A.Vahab

2019-20

SESSION 9 -PLAN

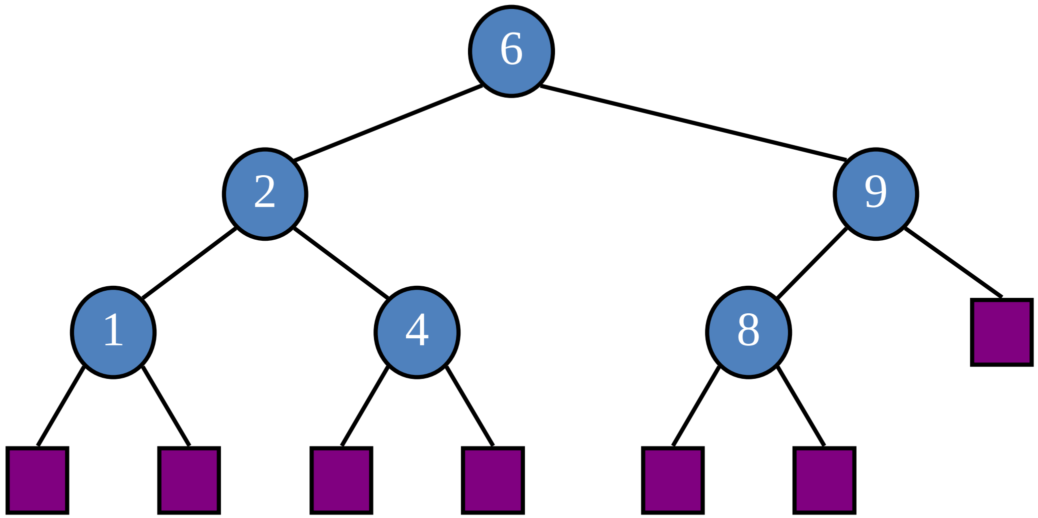
Online Sessions(#)	List of Topic Title	Text/Ref Book/external resource
9	Binary Search Tree - Motivation with the task of Searching and Binary Search Algorithm, Properties of BST, Searching an element in BST, Insertion and Removal of Elements, AVL Trees	T1: 3.1 T1: 3.2

Binary Search Tree

- A binary search tree is a binary tree storing keys (or key-element pairs) at its internal nodes and satisfying the following property
  - Let $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ be three nodes such that $\boldsymbol{u}$ is in the left subtree of $\boldsymbol{v}$ and $\boldsymbol{w}$ is in the right subtree of $\boldsymbol{v}$. We have $\boldsymbol{key}(\boldsymbol{u}) \leq \boldsymbol{key}(\boldsymbol{v}) \leq \boldsymbol{key}(\boldsymbol{w})$

Binary Search Tree

- An inorder traversal of a binary search trees visits the keys in increasing order

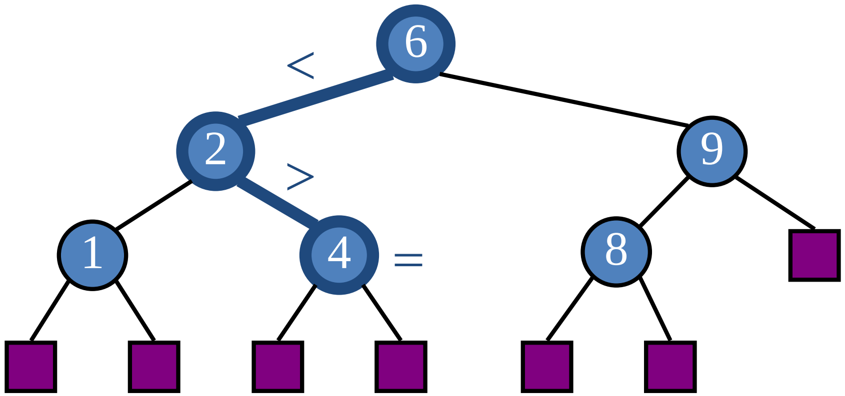


Binary Search Tree- Search

- To search for a key ***k***, we trace a downward path starting at the root
- The next node visited depends on the outcome of the comparison of ***k*** with the key of the current node
- If we reach a leaf, the key is not found and we return NO_SUCH_KEY
- Example: findElement(4)

- External nodes do not store items

Binary Search Tree- Search



Binary Search Tree- Search

Algorithm ***findElement***(*k*, *v*)

- Input: A search key k, and a node v of a binary search tree T
- Output: A node w of the subtree T (v) of T rooted at v, such that either w is an internal node storing key k or w is the external node where an item with key k would belong if it existed

```
if T.isExternal (v)
    return NO_SUCH_KEY
 if k < key(v)
    return findElement(k, T.leftChild(v))
 else if k = key(v)
    return element(v)
 else { k > key(v) }
    return findElement(k, T.rightChild(v))
```

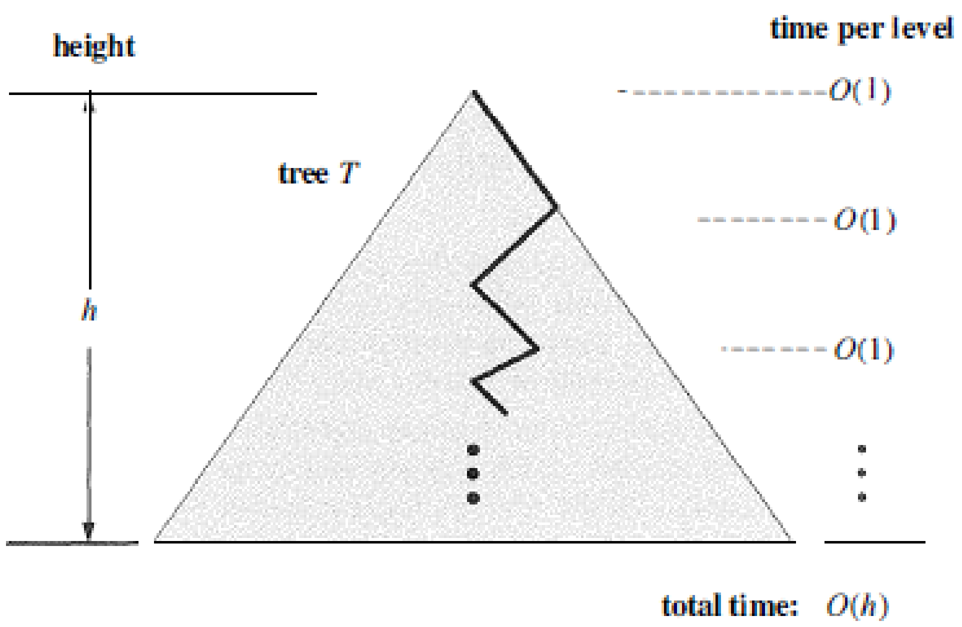
Analysis of Binary Tree Searching

- The binary tree search algorithm executes a constant number of primitive operations for each node it traverses in the tree.
- Each new step in the traversal is made on a child of the previous node.
- That is, the binary tree search algorithm is performed on the nodes of a path of T that starts from the root and goes down one level at a time.
- Thus, the number of such nodes is bounded by h + 1, where h is the height of T.

Analysis of Binary Tree Searching

- In other words, since we spend O(1) time per node encountered in the search, method findElement (or any other standard search operation) runs in O(h) time, where h is the height of the binary search tree T used to implement the dictionary D.
- ie. The running time of searching in a binary search tree T is proportional to the height of T . The height of a tree with n nodes can be **O(log n)**

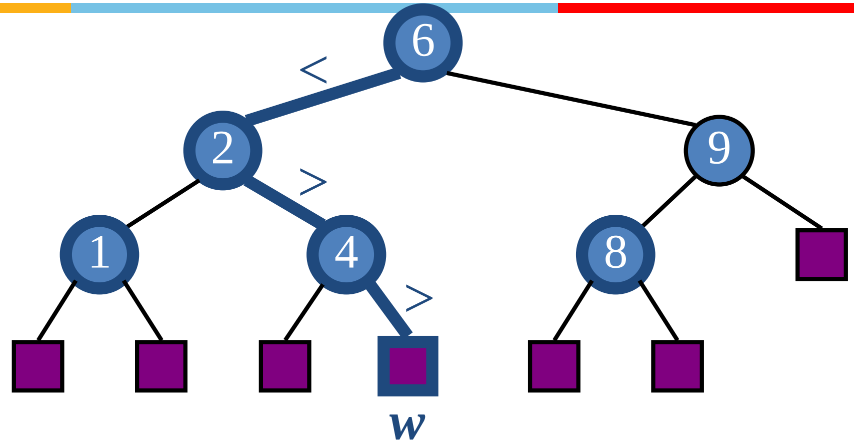
Analysis of Binary Tree Searching

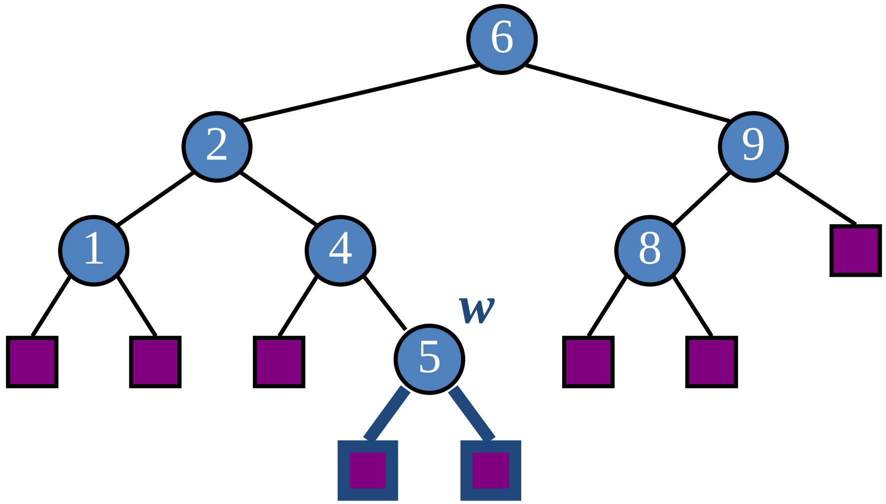


Binary Search Tree- Insertion

- To perform operation insertItem(k, o), we search for key k
- Assume k is not already in the tree, and let w be the leaf reached by the search
- We insert k at node w and expand w into an internal node
- Example: insert 5

Binary Search Tree- Insertion



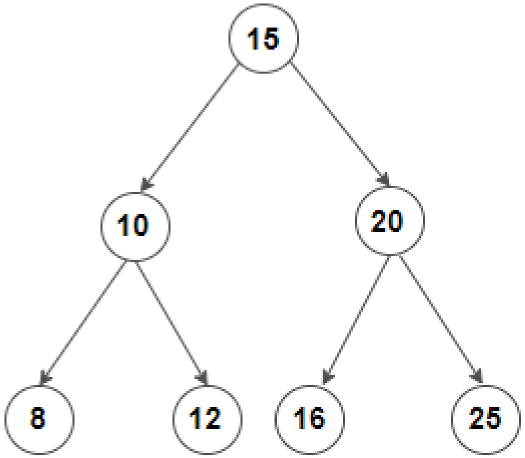


In-order Successor and Predecessor

- In a Binary Search Tree, the successor of a given key is the smallest number which is larger than the key.
- In the same way, a predecessor is the largest number which is smaller than the key.
- If X has two children then its in-order predecessor is the maximum value in its left subtree and its in-order successor the minimum value in its right subtree.

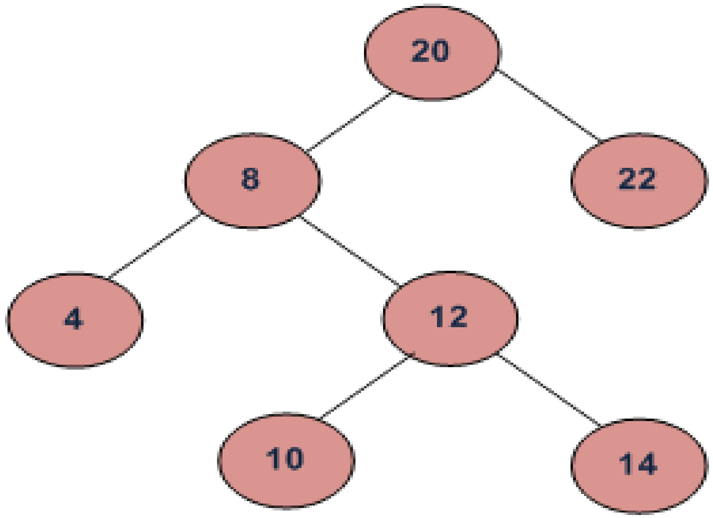
Predecessor

- Inorder predecessor of 8 doesn't exist
- Inorder predecessor of 20 is 16
- Inorder predecessor of 12 is 10



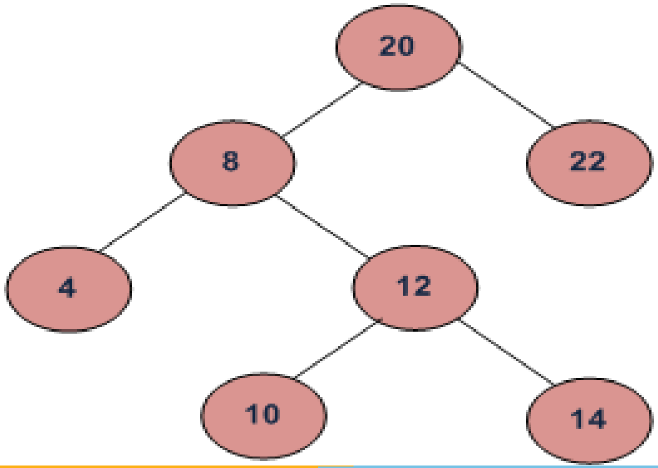
Predecessor

- Inorder predecessor of a node is a node with maximum value in its left subtree. i.e left subtree's right most child.
- If left subtree doesn't exists, then predecessor is one of the ancestors. Travel up using the parent pointer until you see a node which is right child of it's parent. The parent of such a node is the predecessor.



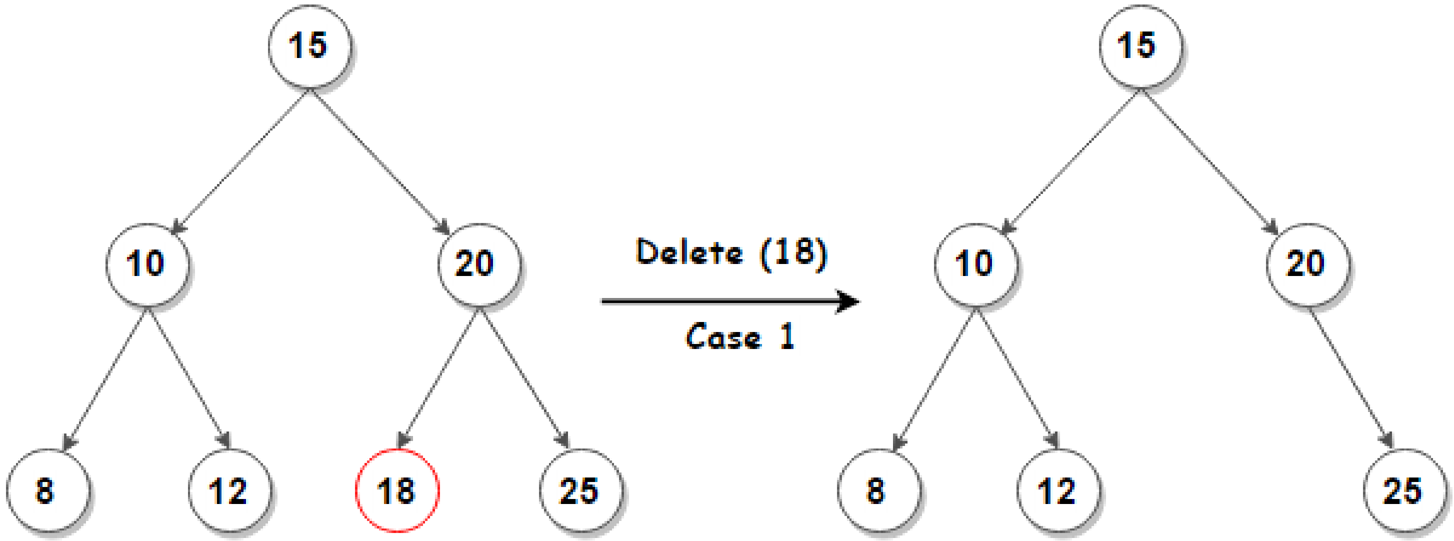
Successor

- If right subtree of node is not NULL, then succ lies in right subtree.Go to right subtree and return the node with minimum key value in right subtree.i.e **right subtree's left most child**.
- If right subtree of node is NULL, then succ is one of the ancestors. Travel up using the parent pointer until you see a node which is left child of it's parent. The parent of such a node is the succ.



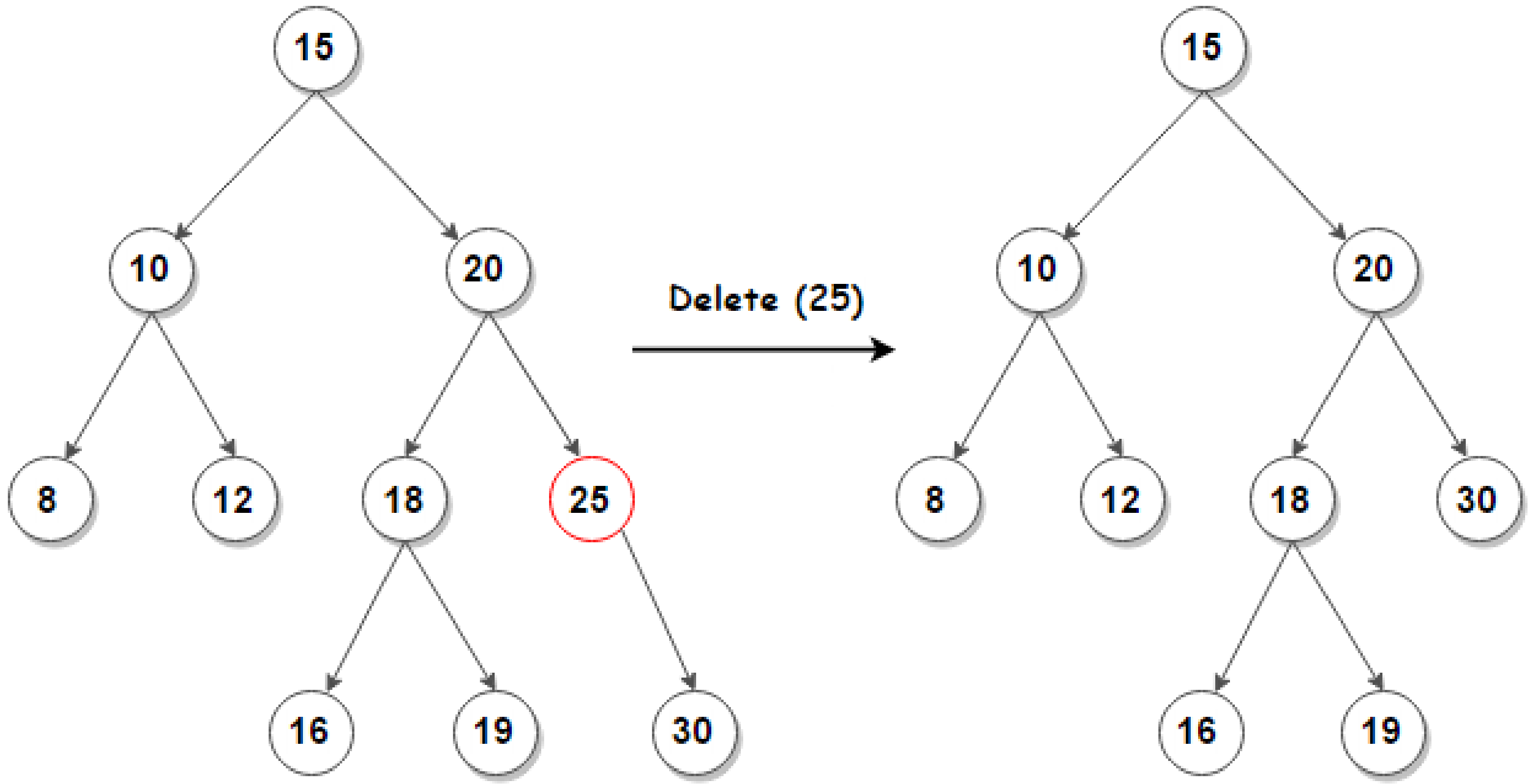
Binary Search Tree-Deletion

- Deleting a node with no Children

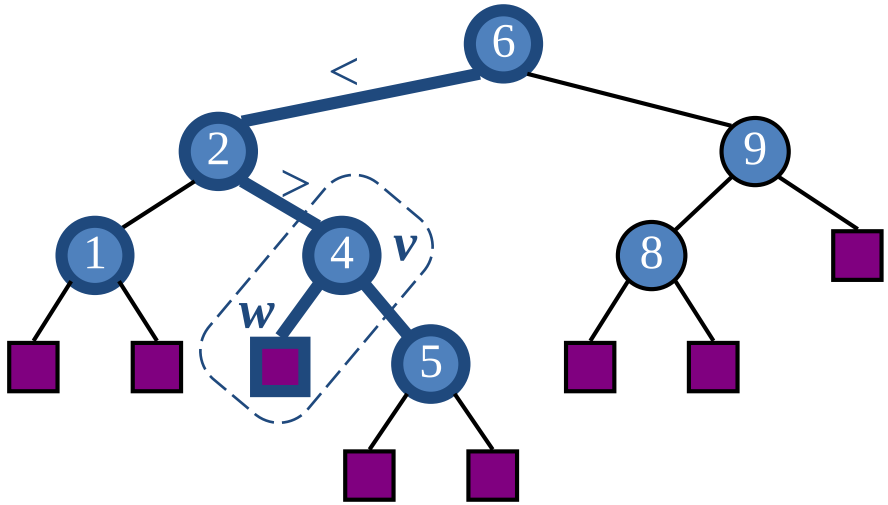


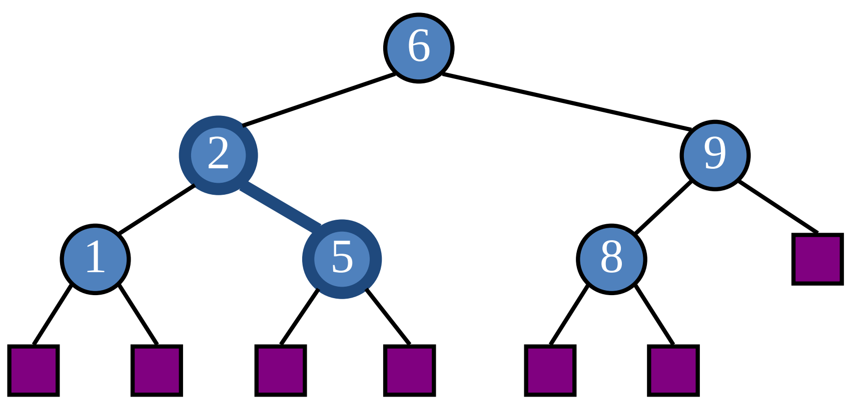
Binary Search Tree-Deletion

- **Deleting a node with 1 child**: Remove the node and replace it with its child



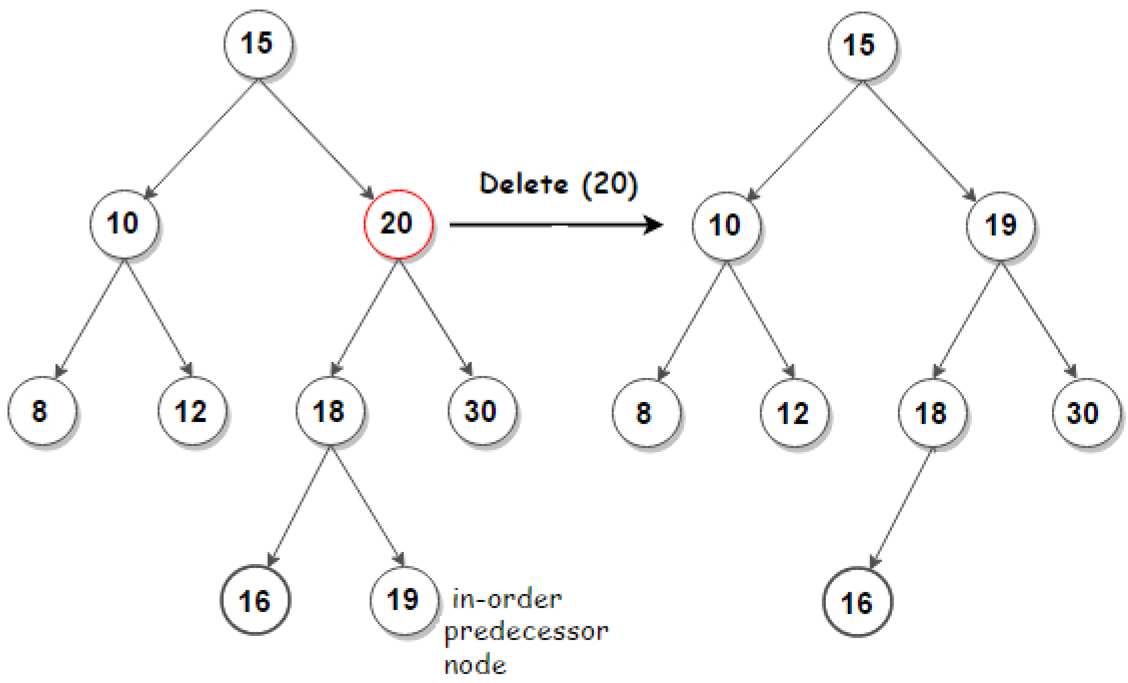
Binary Search Tree-Deletion





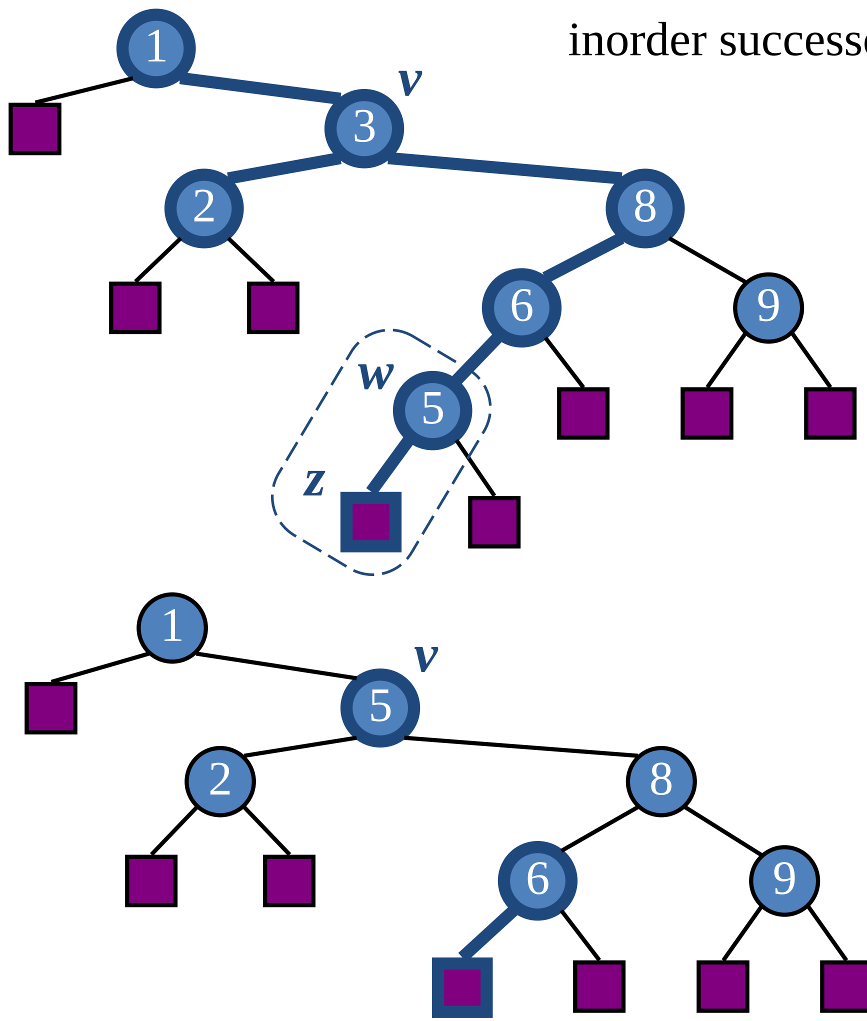
Binary Search Tree-Deletion

- Deleting a node with 2 children: Replace the node with its inorder successor(Predecessor)



Binary Search Tree-Deletion

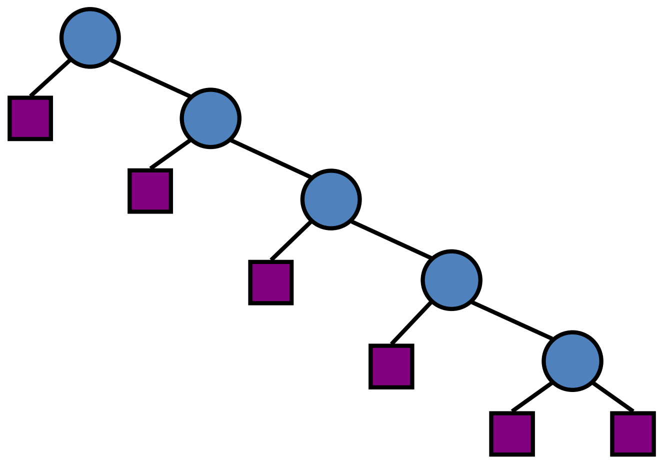
Delete node 3: Replace the node with its inorder successor ,node 5

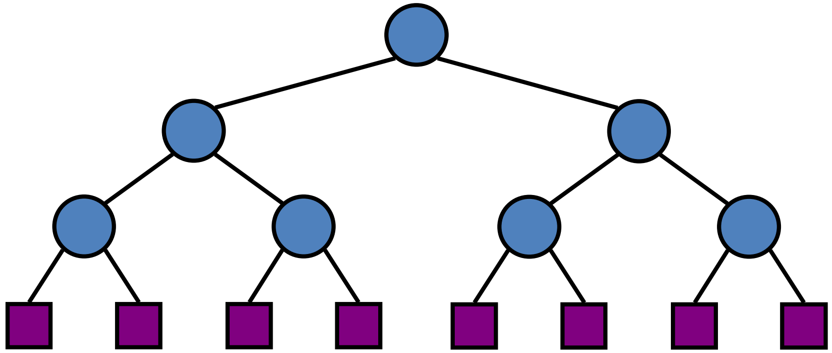


Performance

- Consider a BST with n items and height h
- The space used is O(n)
- Methods findElement, insertItem and removeElement take O(h) time
- The height h is O(n) in the worst case and O(log n) in the best case

Binary Search Tree





Balanced tree

- The worst-case performance, a BST achieves for various operations is linear time, which is no better than the performance of sequence-based dictionary implementations (such as log files and lookup tables).
- A simple way of correcting this problem is balanced binary search tree.

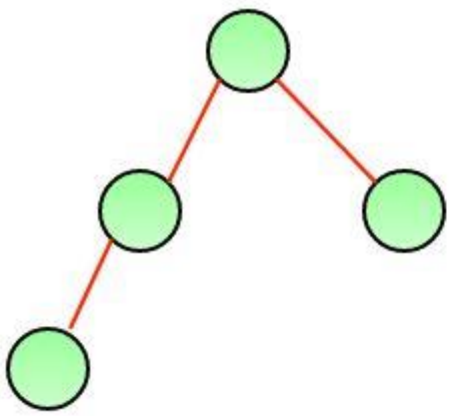
Balanced tree

- A balanced tree is a tree where every leaf is “not more than a certain distance” away from the root than any other leaf.
- Add a rule to the binary search tree definition that will maintain a logarithmic height for the tree
- Height-balance property

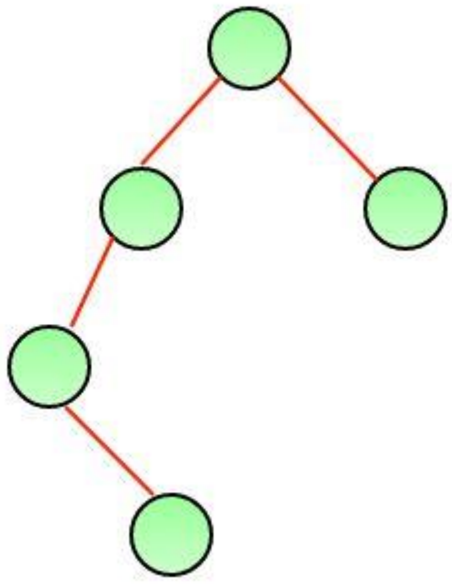
Balanced tree

- Height-Balance Property:

For every internal node v of T, the heights of the children of v can differ by at most 1 .



A height balanced tree



Not a height balanced tree

THANK YOU!!!

BITS Pilani
Hyderabad Campus