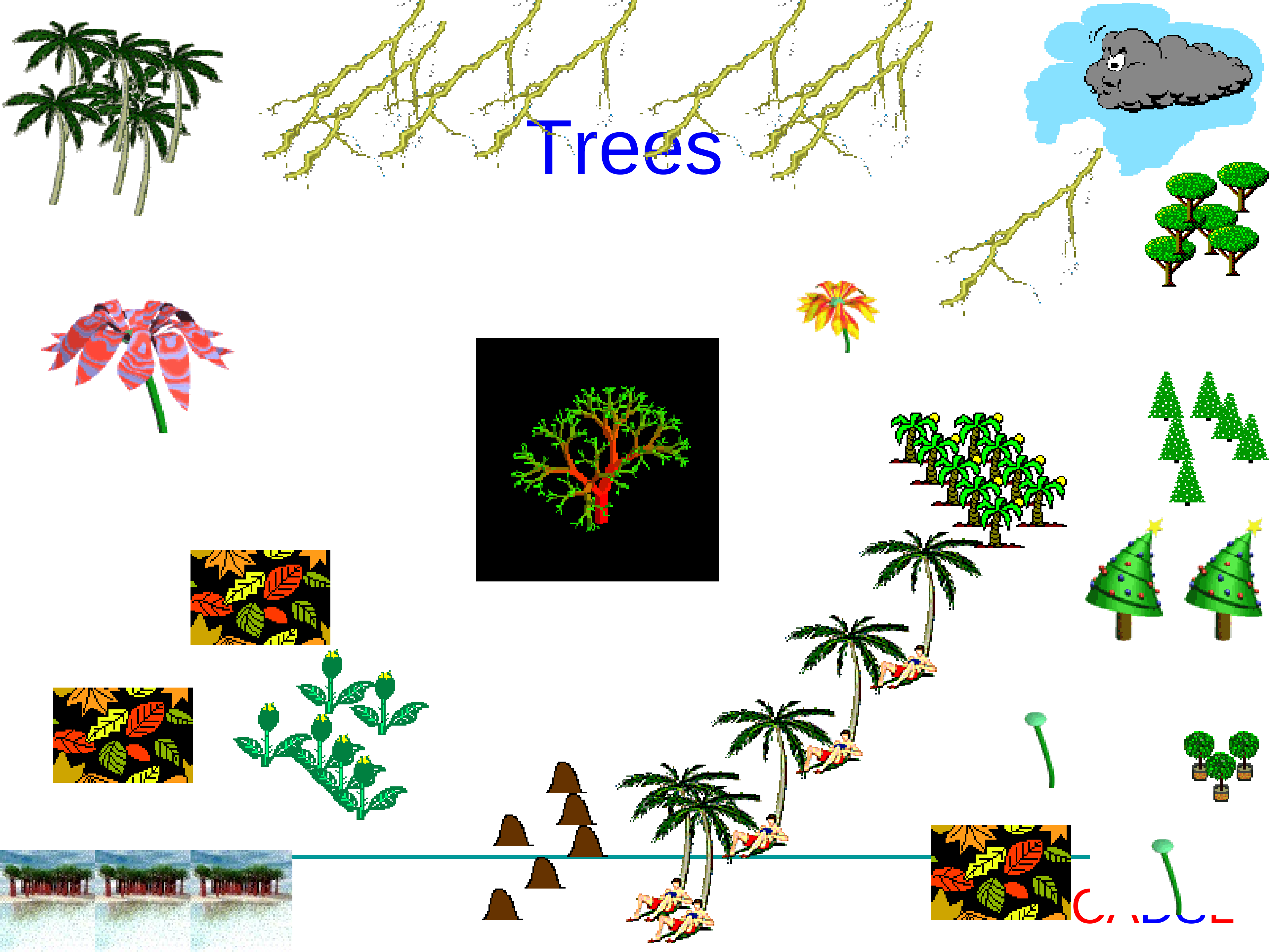
Data Structures

Tree

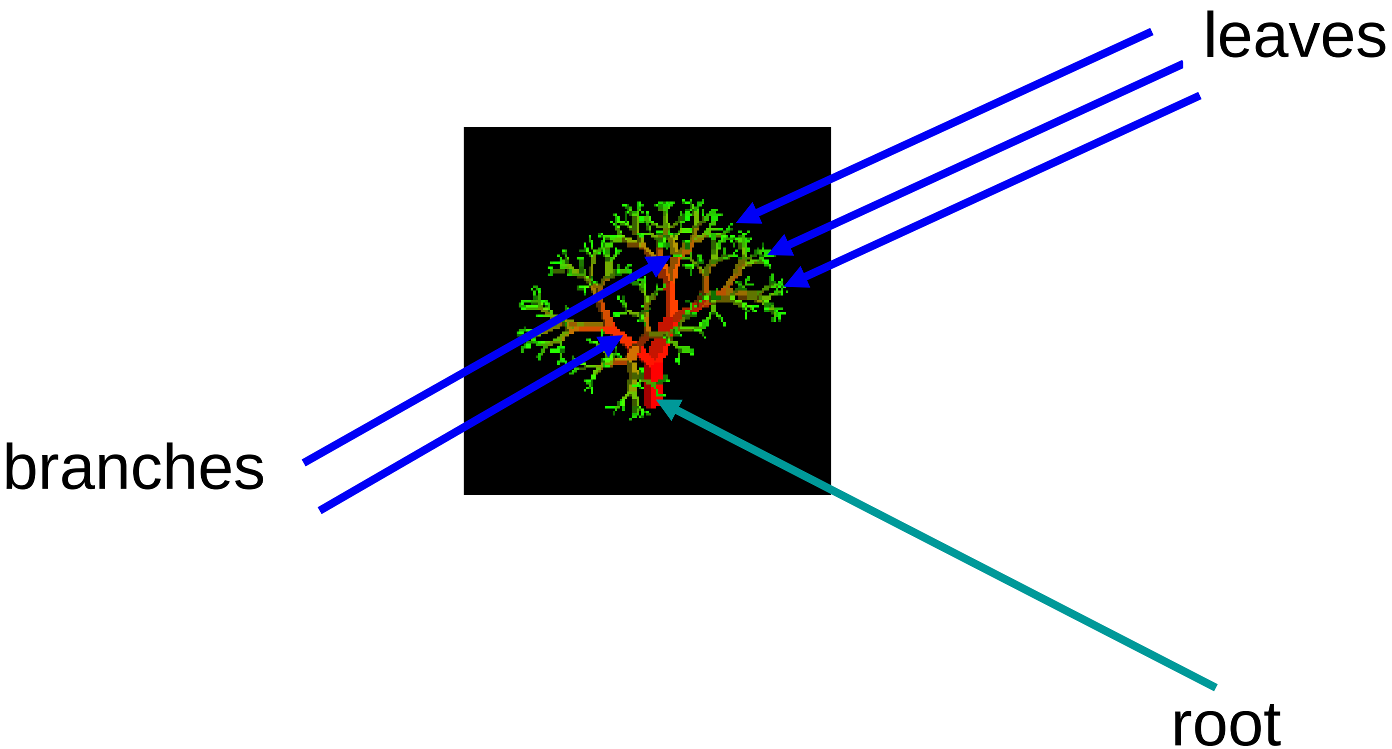
Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

EE-717/453:Advance Computing for Electrical Engineers

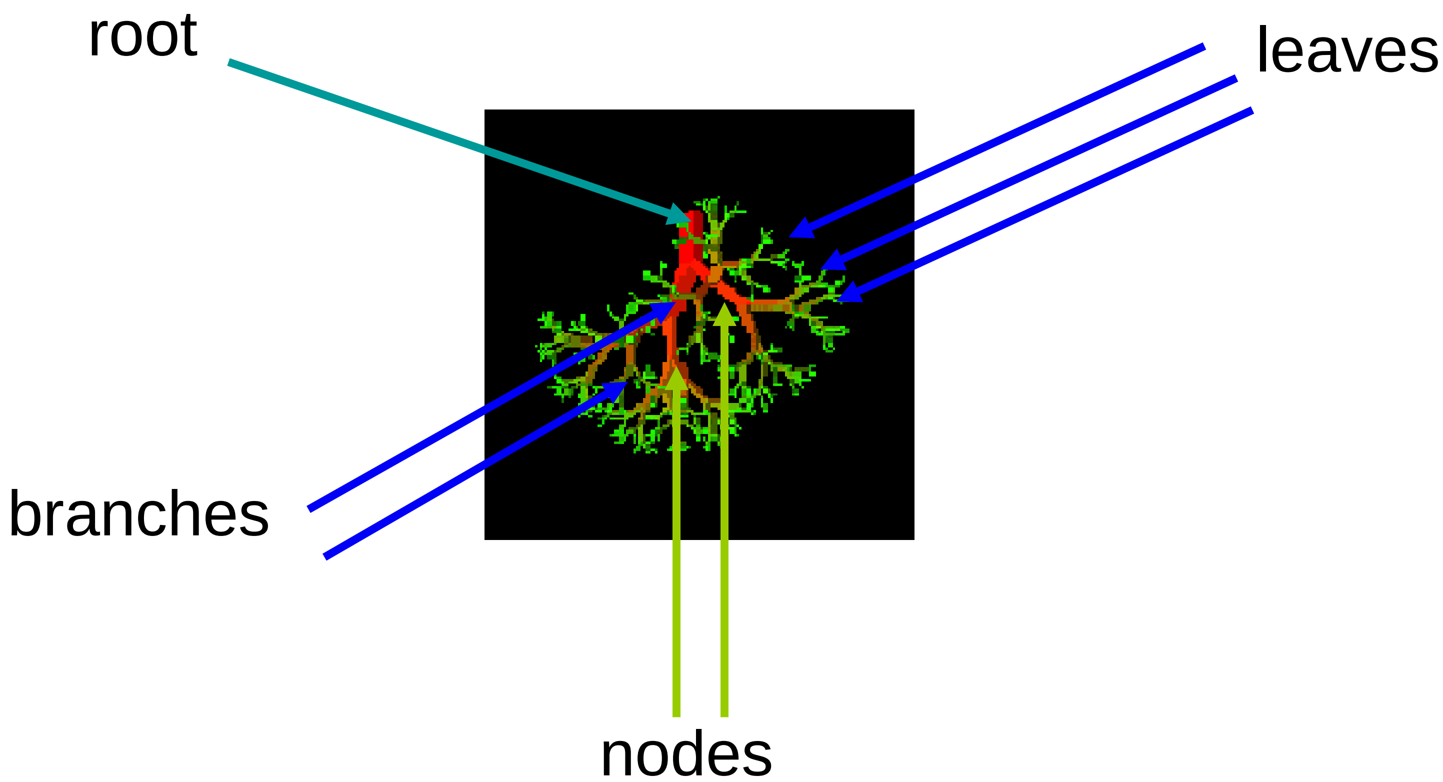
Lecture 6 (08 Aug 2013)


Trees

Nature Lover's View of A Tree



Computer Scientist's View



Linear Lists And Trees

- Linear lists are useful for serially ordered data.
  - (e0, e1, e2, …, en-1)
  - Days of week.
  - Months in a year.
  - Students in this class.
- Trees are useful for hierarchically ordered data.
  - Employees of a corporation.
    - President, vice presidents, managers, and so on.
  - Java's classes.
    - Object is at the top of the hierarchy.
    - Subclasses of Object are next, and so on.

Hierarchical Data And Trees

- The element at the top of the hierarchy is the root.
- Elements next in the hierarchy are the children of the root.
- Elements next in the hierarchy are the and children of the root, and so on.
- Elements that have no children are

.

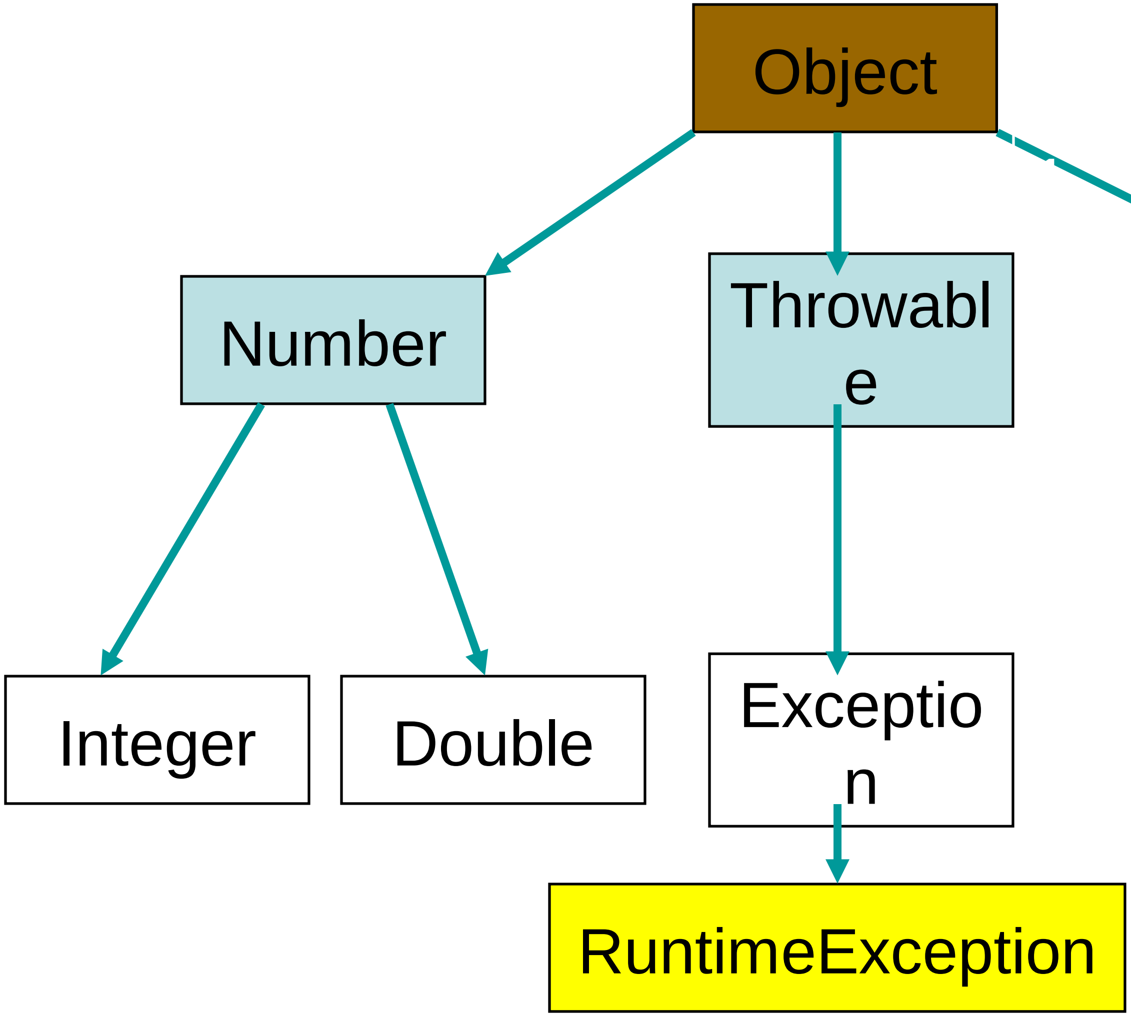
Definition

- A tree   is a finite nonempty set of elements.
- One of these elements is called the root.
- The remaining elements, if any, are partitioned into trees, which are called the subtrees of t.

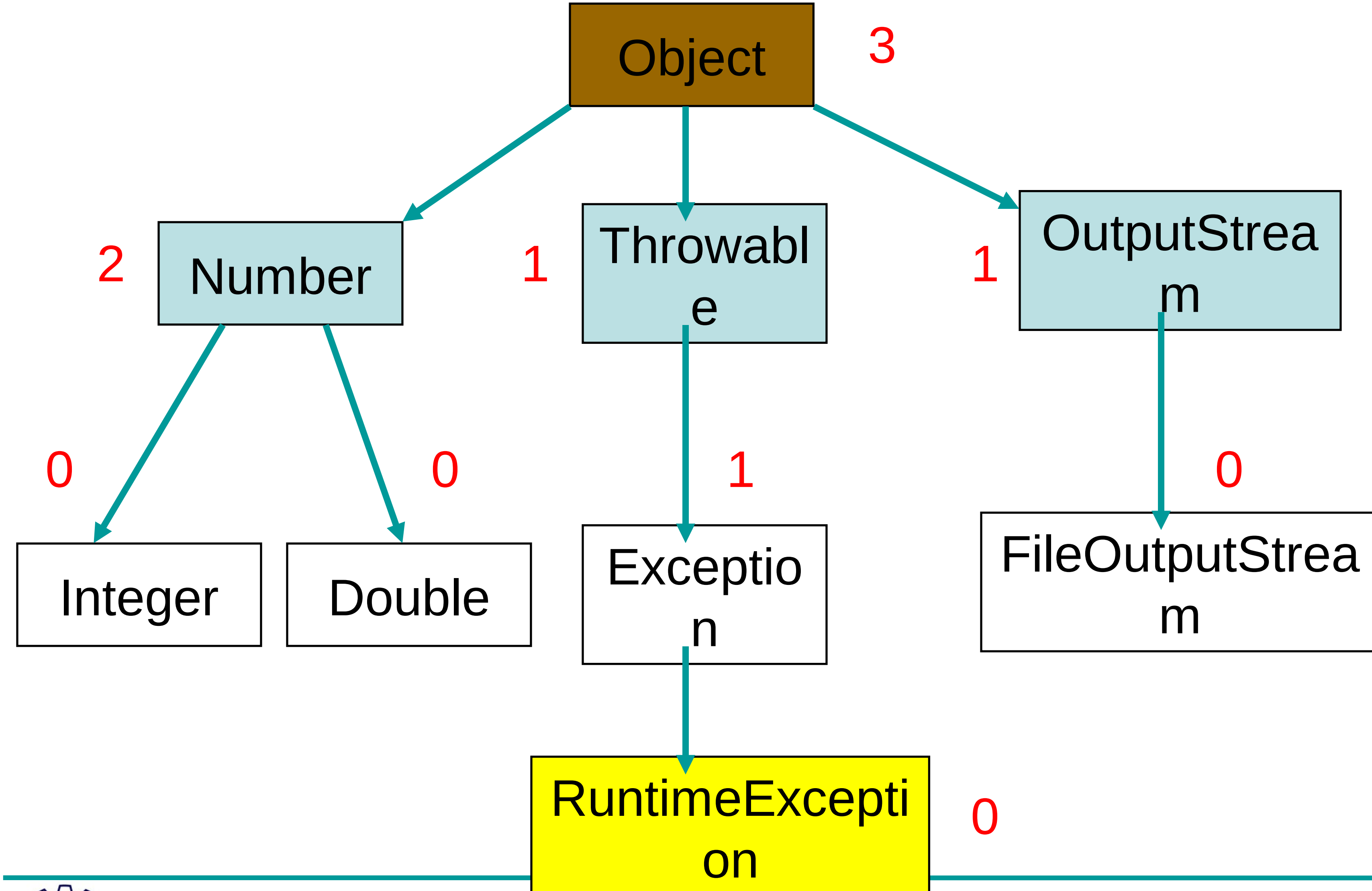
Caution

- Some texts start level numbers at 0 rather than at 1.
- Root is at level 0.
- Its children are at level 1.
- The grand children of the root are at level 2.
- And so on.
- We shall number levels with the root at level 1.

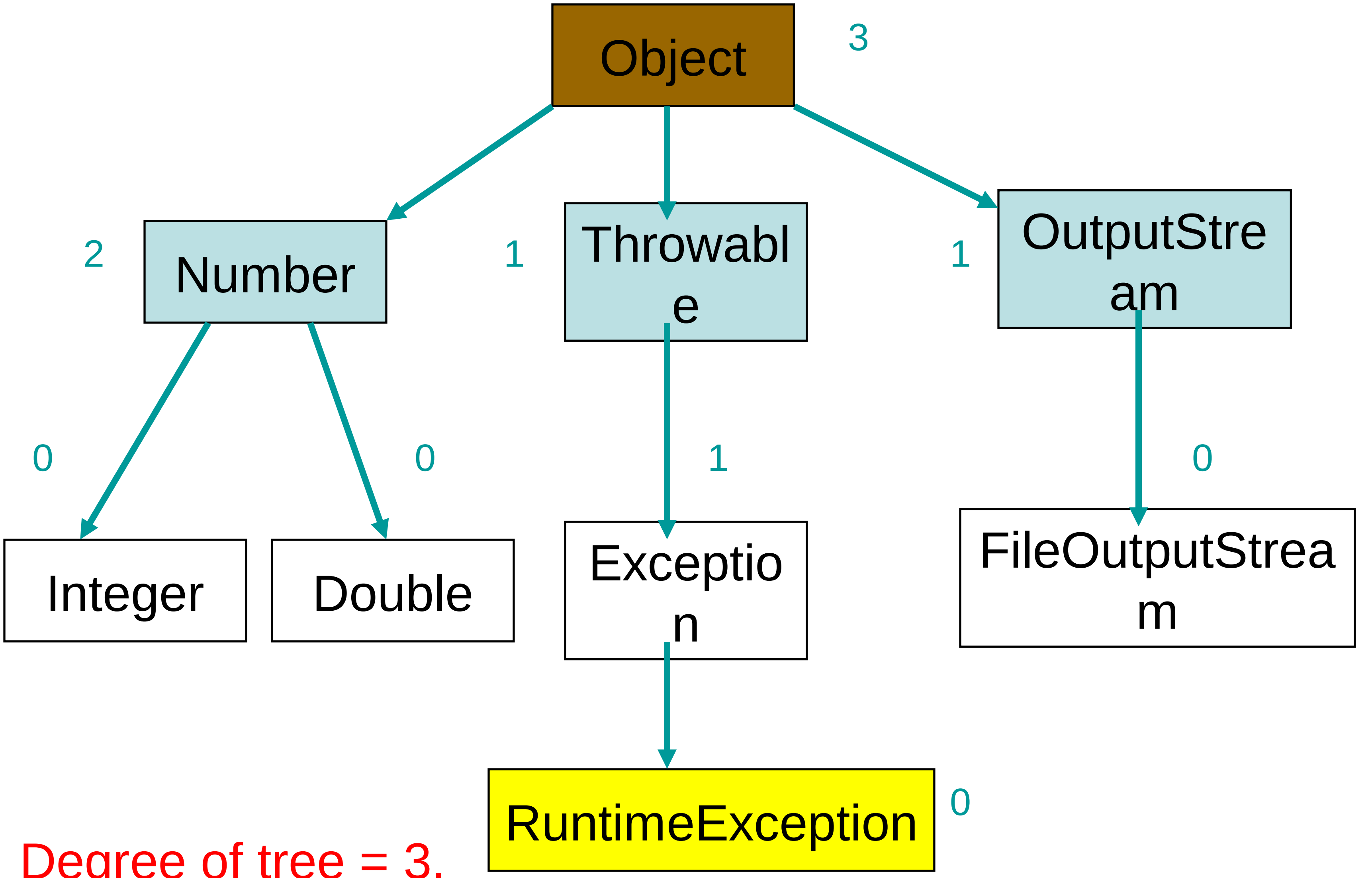
height = depth = number of levels



Node Degree = Number Of Children



Tree Degree = Max Node Degree



Degree of tree = 3.

Binary Tree

- Finite (possibly empty) collection of elements.
- A nonempty binary tree has a root element.
- The remaining elements (if any) are partitioned into two binary trees.
- These are called the left and right subtrees of the binary tree.

Differences Between A Tree & A Binary Tree

- No node in a binary tree may have a degree more than 2, whereas there is no limit on the degree of a node in a tree.
- A binary tree may be empty; a tree cannot be empty.

Differences Between A Tree & A Binary Tree

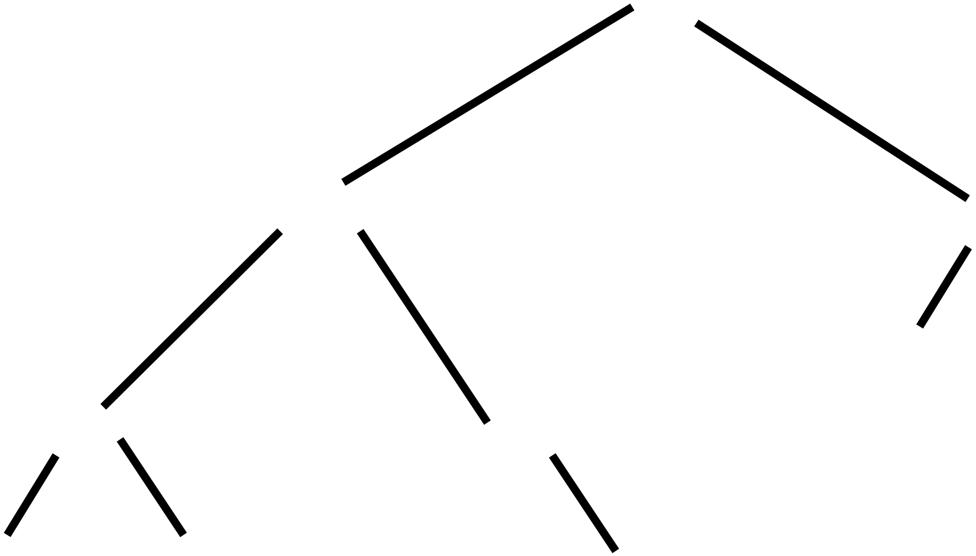
- The subtrees of a binary tree are ordered; those of a tree are not ordered.

- Are different when viewed as binary trees.
- Are the same when viewed as trees.

Arithmetic Expressions

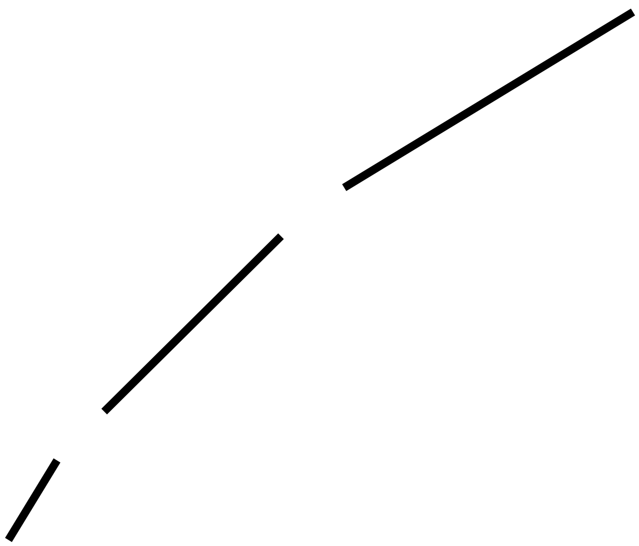
- (a + b) * (c + d) + e – f/g*h + 3.25
- Expressions comprise three kinds of entities.
  - Operators (+, -, /, *).
  - Operands (a, b, c, d, e, f, g, h, 3.25, (a + b), (c + d), etc.).
  - Delimiters ((,)).

Binary Tree Properties & Representation



Minimum Number Of Nodes

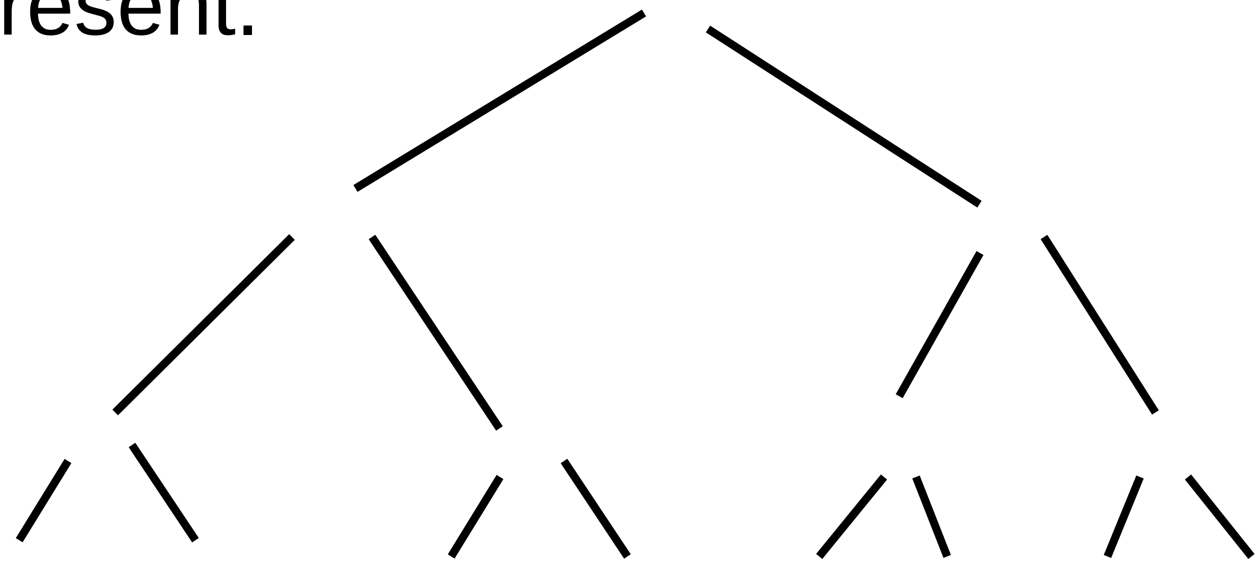
- Minimum number of nodes in a binary tree whose height is h.
- At least one node at each of first h levels.



minimum number of nodes is h

Maximum Number Of Nodes

- All possible nodes at first h levels are present.



Maximum number of nodes

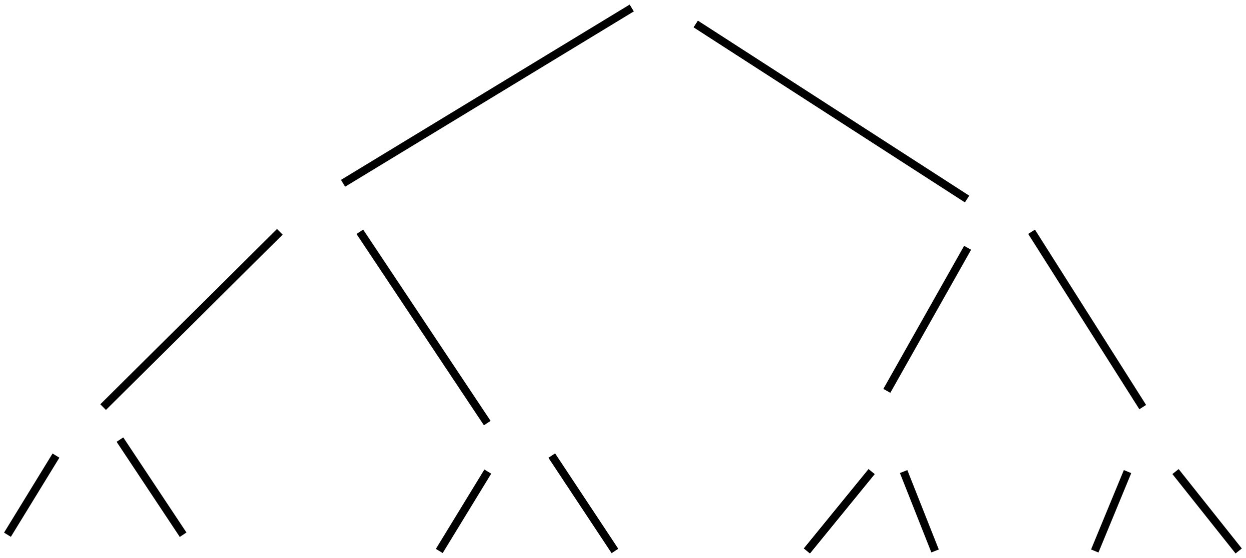
= 1 + 2 + 4 + 8 + … + 2h-1 = 2h - 1

Number Of Nodes & Height

- Let n be the number of nodes in a binary tree whose height is h.
- h <= n <= 2h – 1
- log2(n+1) <= h <= n

Full Binary Tree

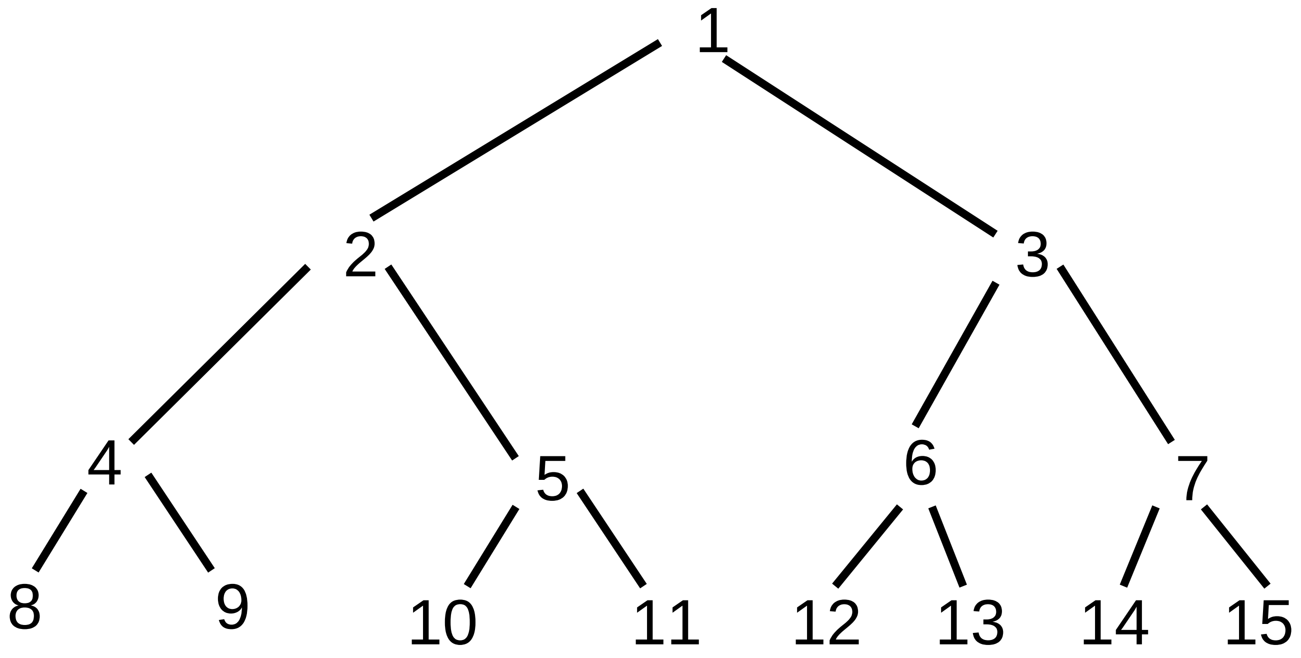
- A full binary tree of a given height h has 2h – 1 nodes.



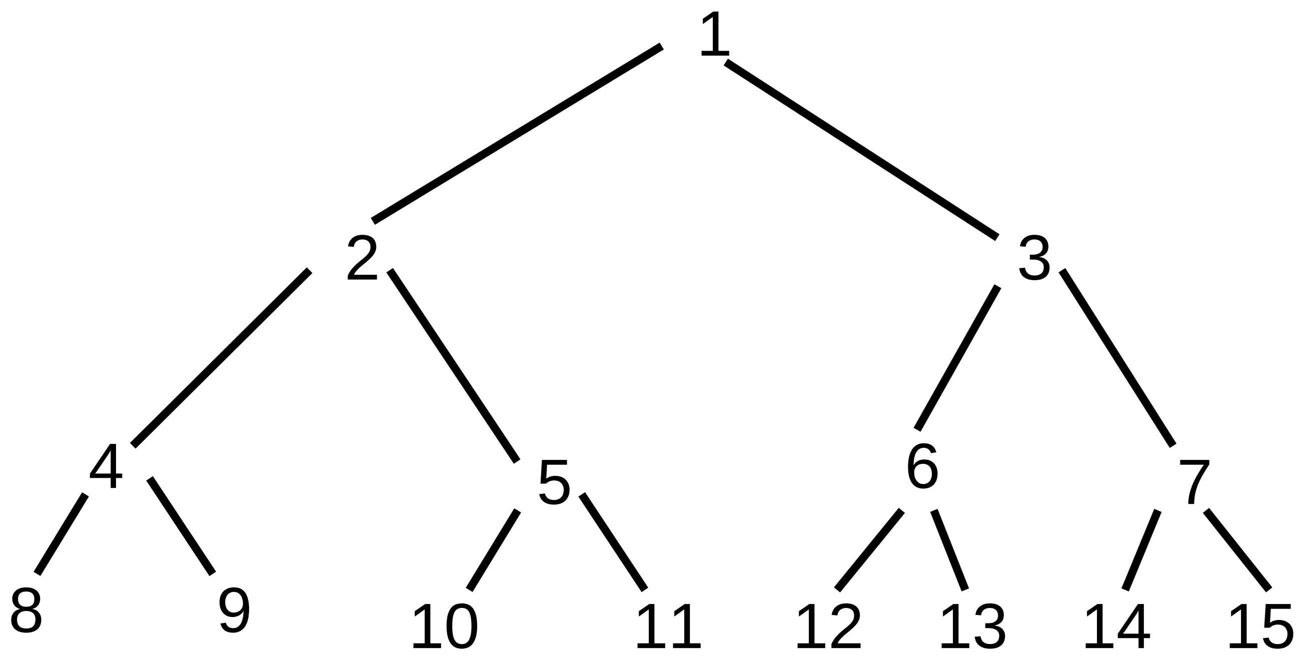
Height 4 full binary tree.

Numbering Nodes In A Full Binary Tree

- Number the nodes 1 through 2h – 1.
- Number by levels from top to bottom.
- Within a level number from left to right.

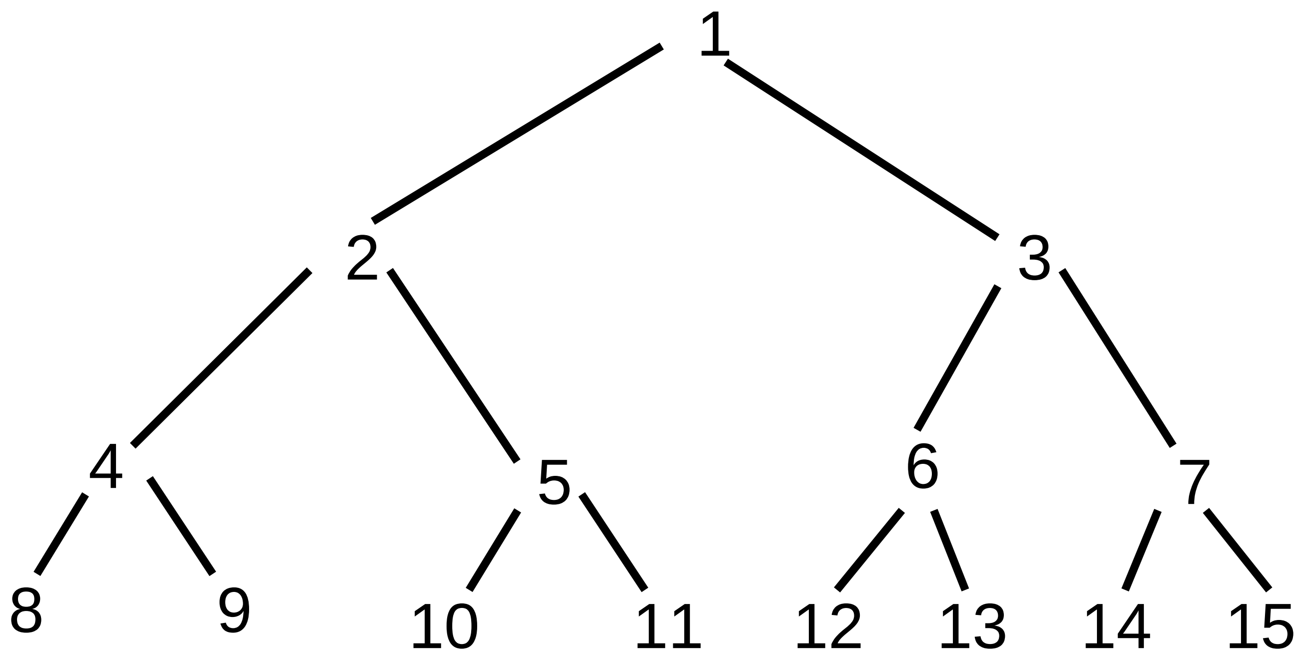


Node Number Properties



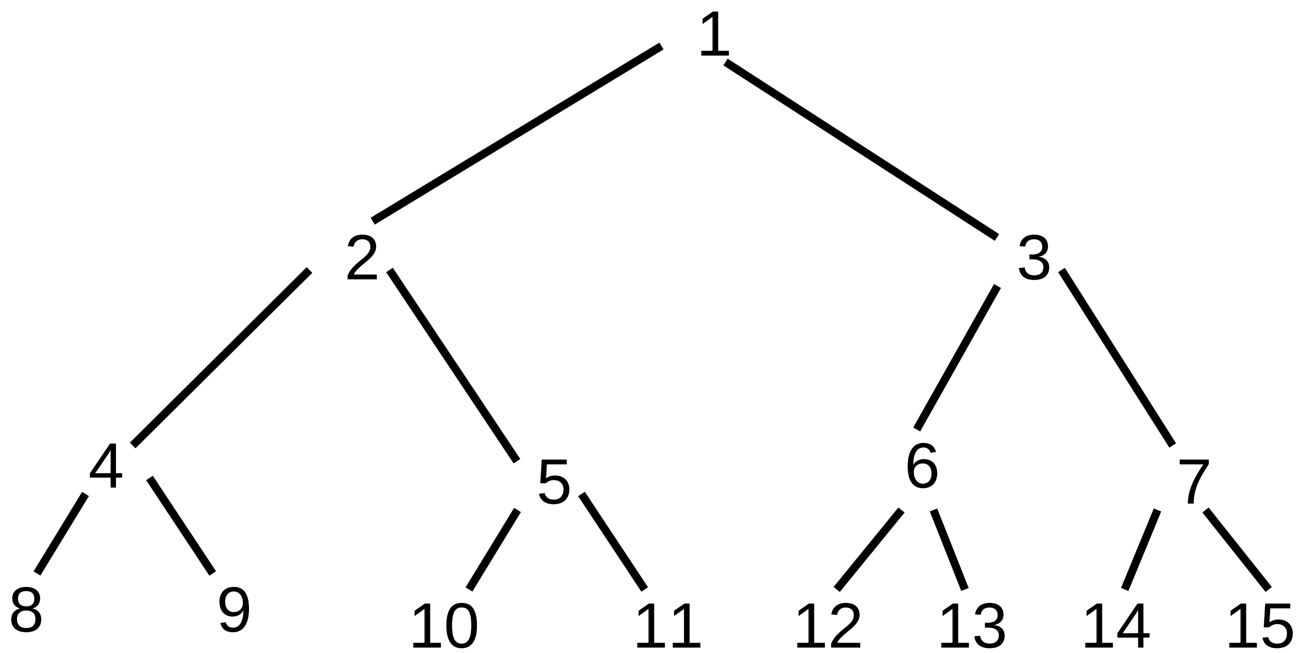
- Parent of node i is node i / 2, unless i = 1.
- Node 1 is the root and has no parent.

Node Number Properties



- Left child of node i is node 2i, unless 2i > n, where n is the number of nodes.
- If 2i > n, node i has no left child.

Node Number Properties

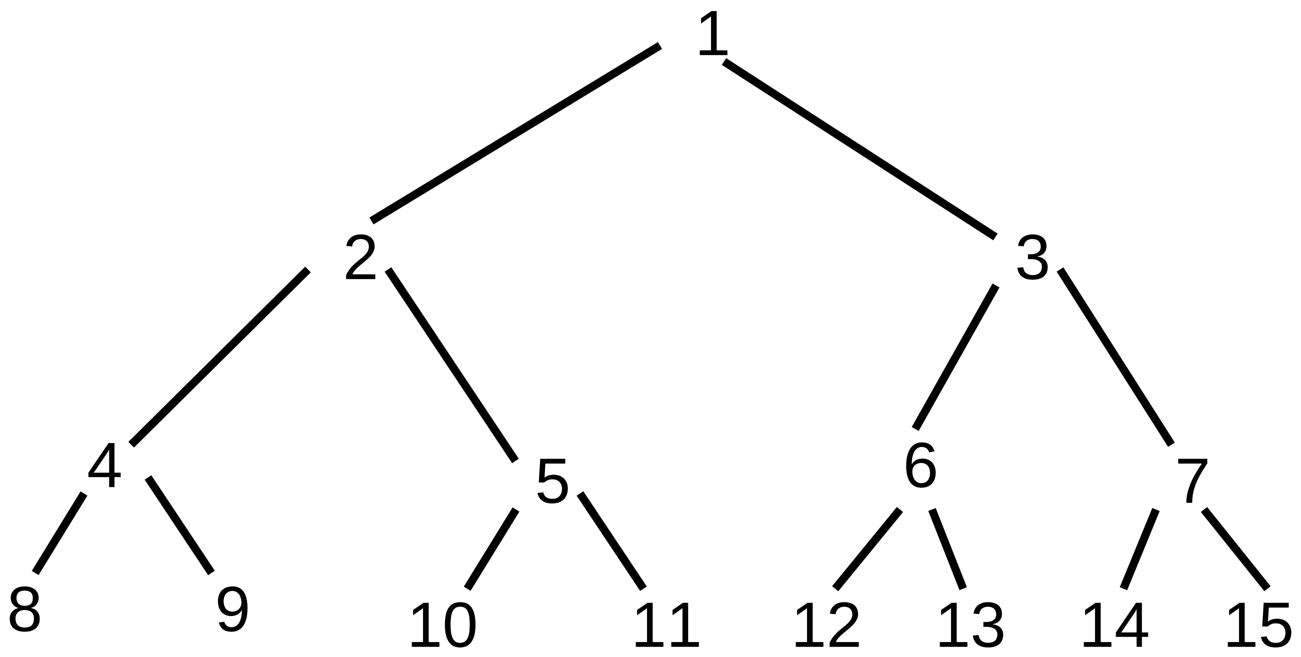


- Right child of node i is node 2i+1, unless 2i+1 > n, where n is the number of nodes.
- If 2i+1 > n, node i has no right child.

Complete Binary Tree With n Nodes

- Start with a full binary tree that has at least n nodes.
- Number the nodes as described earlier.
- The binary tree defined by the nodes numbered 1 through n is the unique n node complete binary tree.

Example



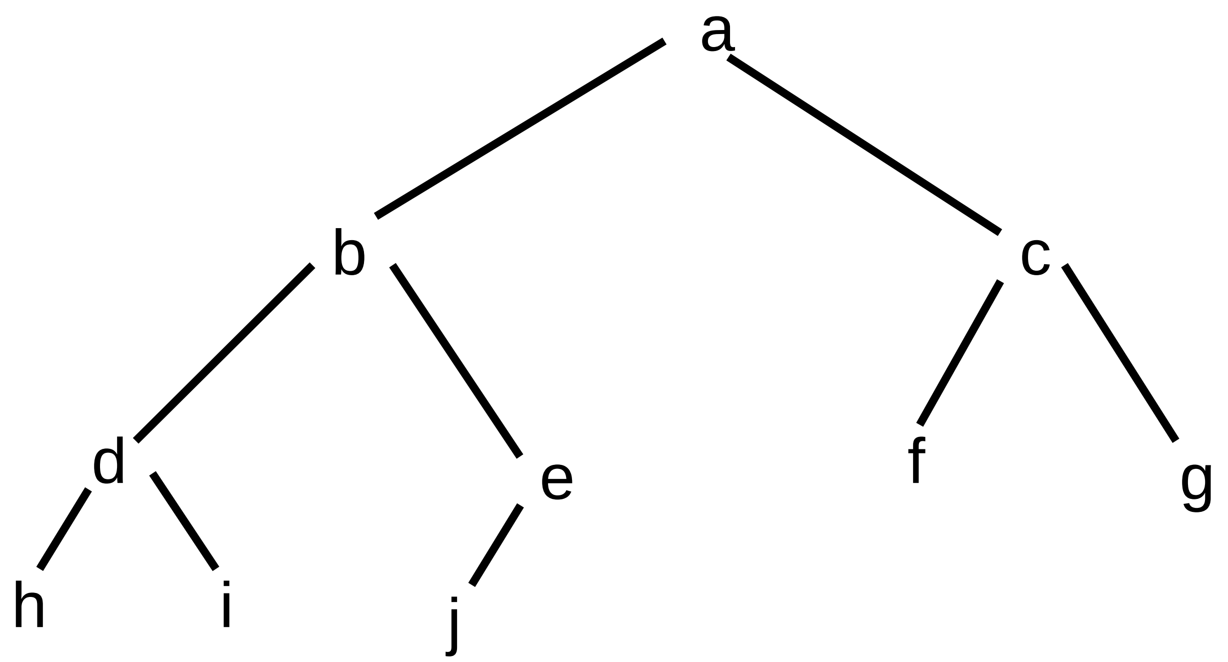
- Complete binary tree with 10 nodes.

Binary Tree Representation

- Array representation.
- Linked representation.

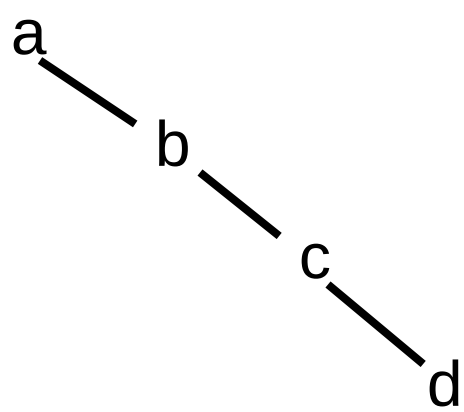
Array Representation

- Number the nodes using the numbering scheme for a full binary tree. The node that is numbered i is stored in tree[i].



tree[] a b c d e f g h i j

Right-Skewed Binary Tree



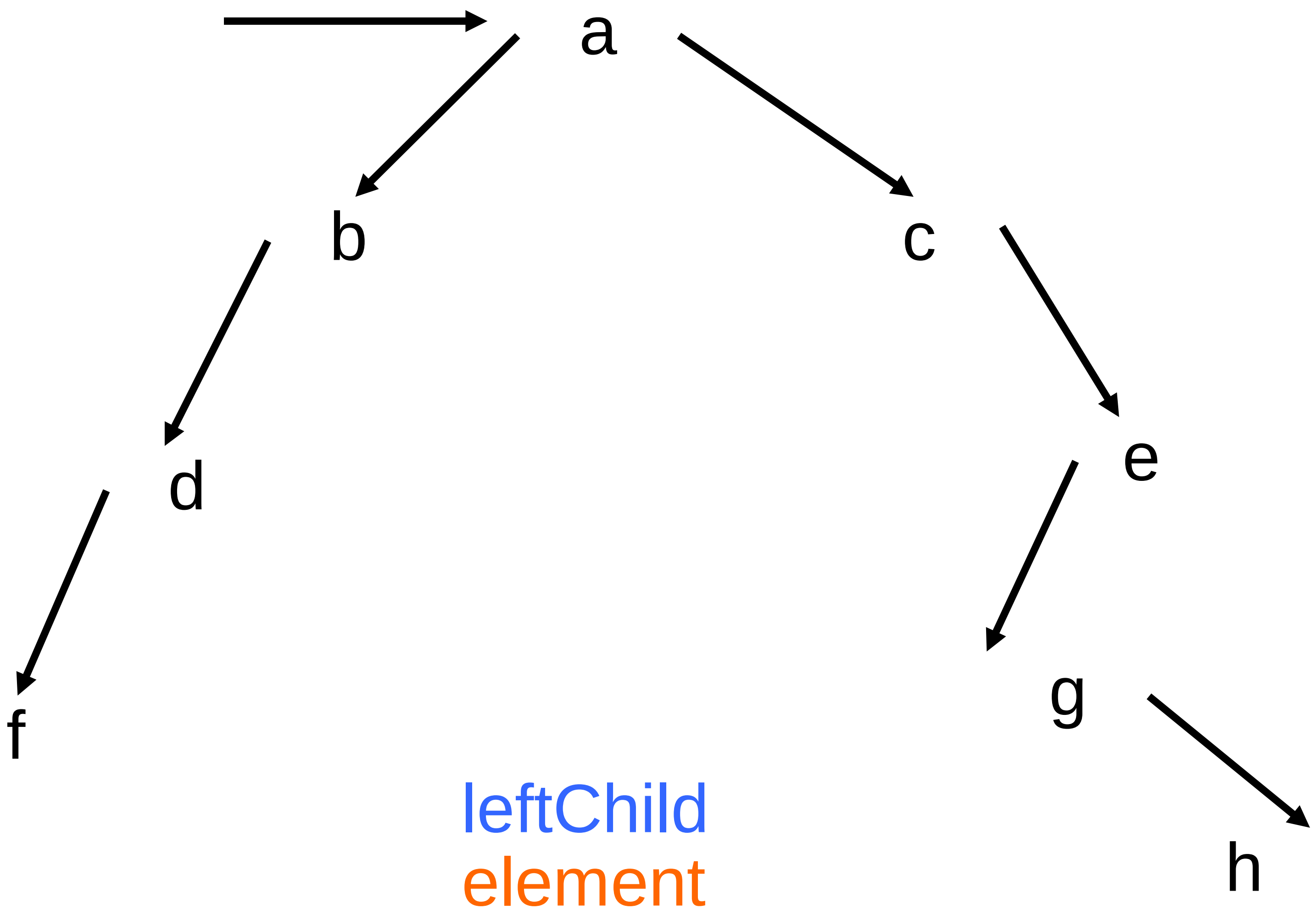
tree[] a - b - - - c - - - - - - - d

- An n node binary tree needs an array whose length is between n+1 and 2n.

Linked Representation

- Each binary tree node is represented as an object whose data type is BinaryTreeNode.
- The space required by an n node binary tree is n * (space required by one node).

Linked Representation Example



leftChild
element
rightChild

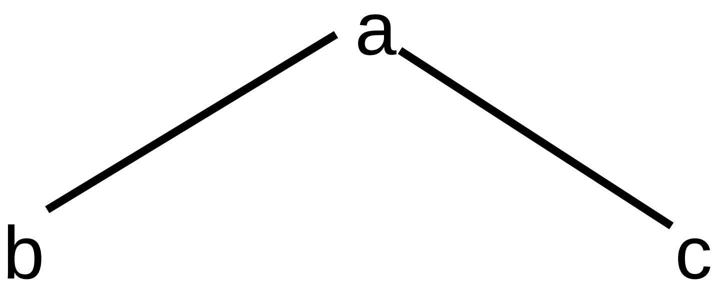
Binary Tree Traversal

- Many binary tree operations are done by performing a traversal of the binary tree.
- In a traversal, each element of the binary tree is visited exactly once.
- During the visit of an element, all action (make a clone, display, evaluate the operator, etc.) with respect to this element is taken.

Binary Tree Traversal Methods

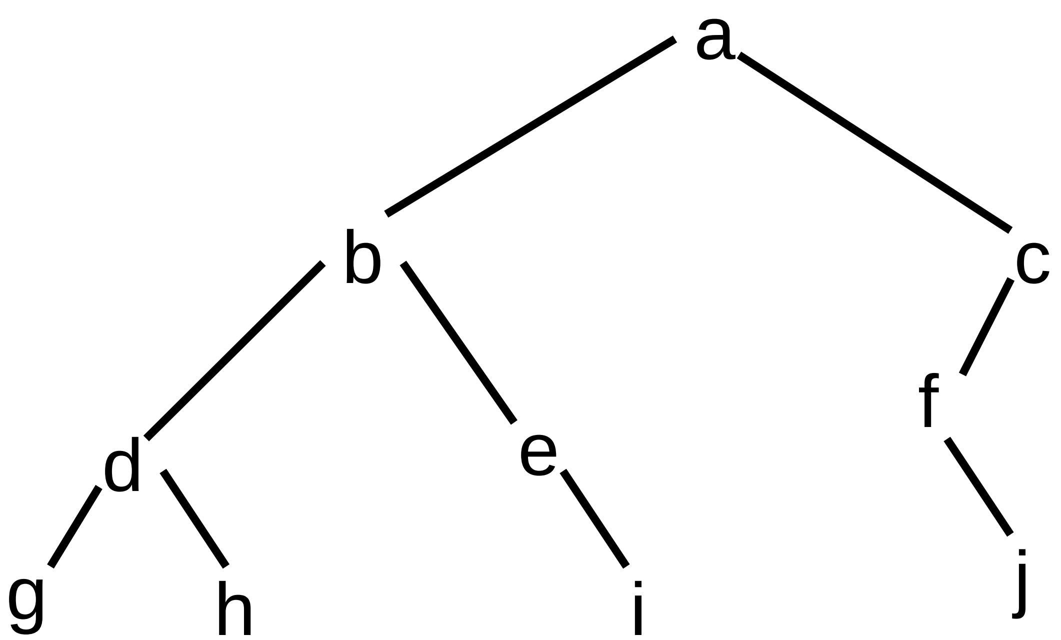
- Preorder
- Inorder
- Postorder
- Level order

Preorder Example (visit = print)



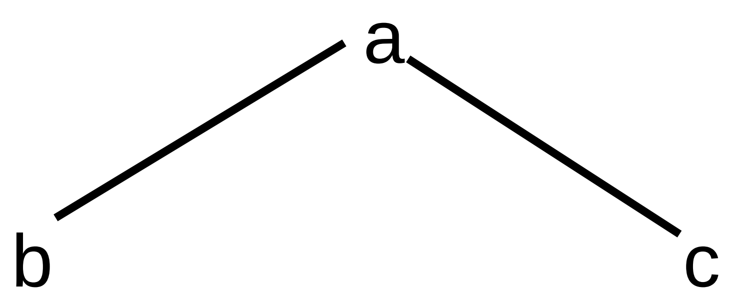
a b c

Preorder Example (visit = print)



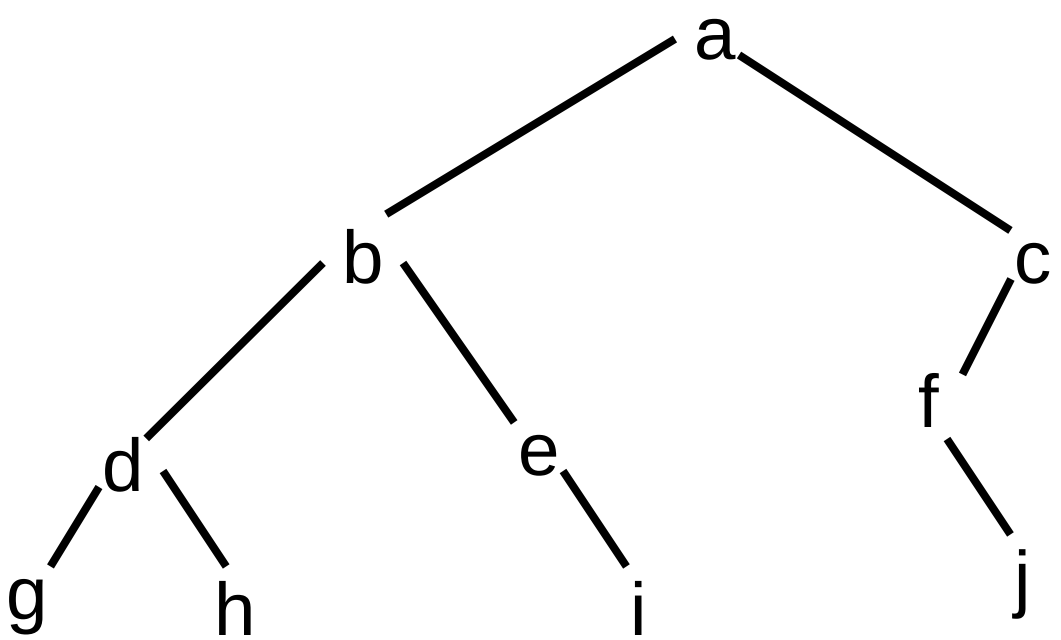
a b d g h e i c f j

Inorder Example (visit = print)



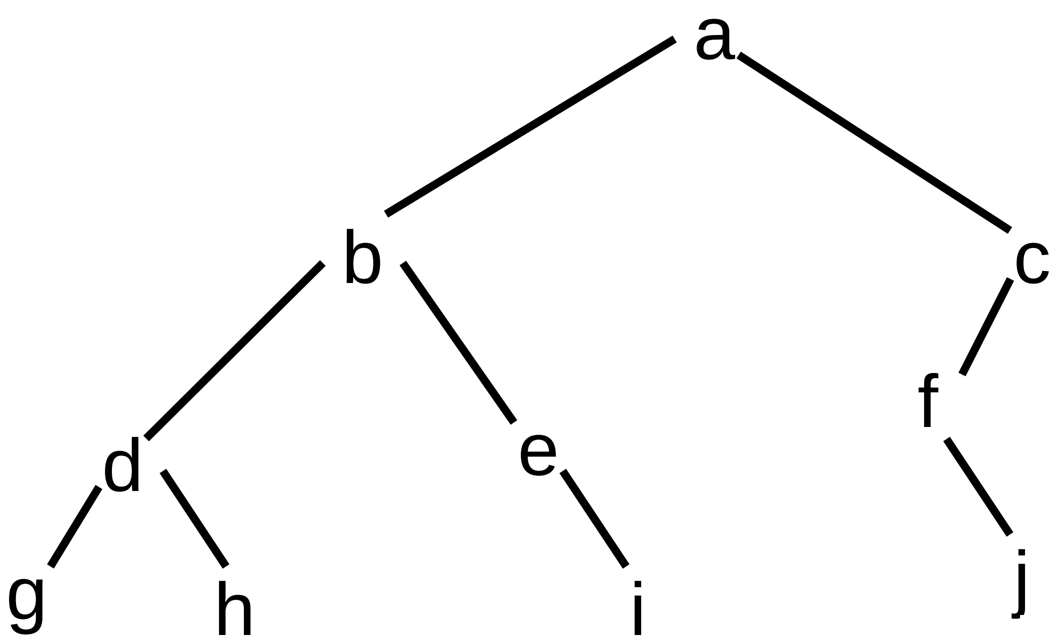
b a c

Inorder Example (visit = print)



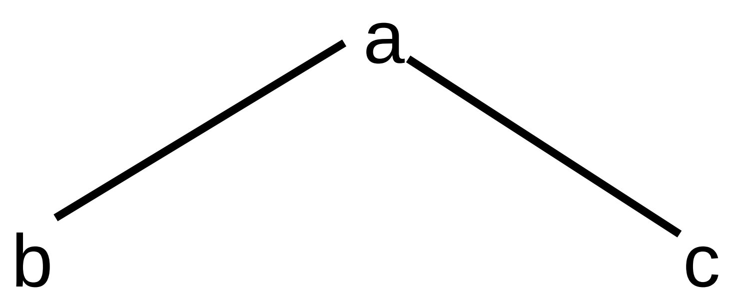
g d h b e i a f j c

Inorder By Projection (Squishing)



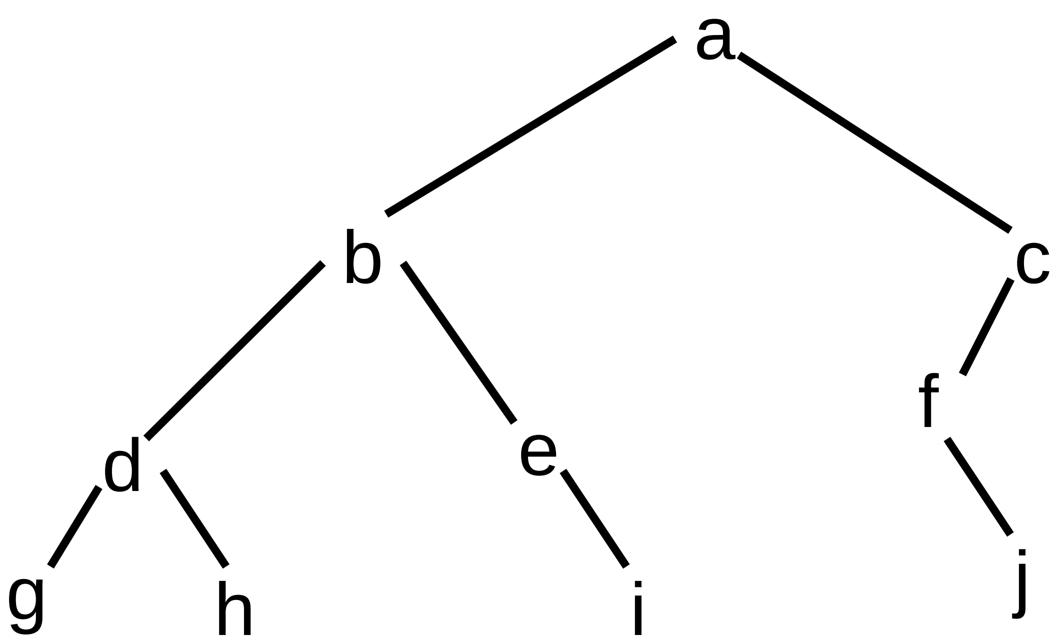
g d h b e i a f j c

Postorder Example (visit = print)



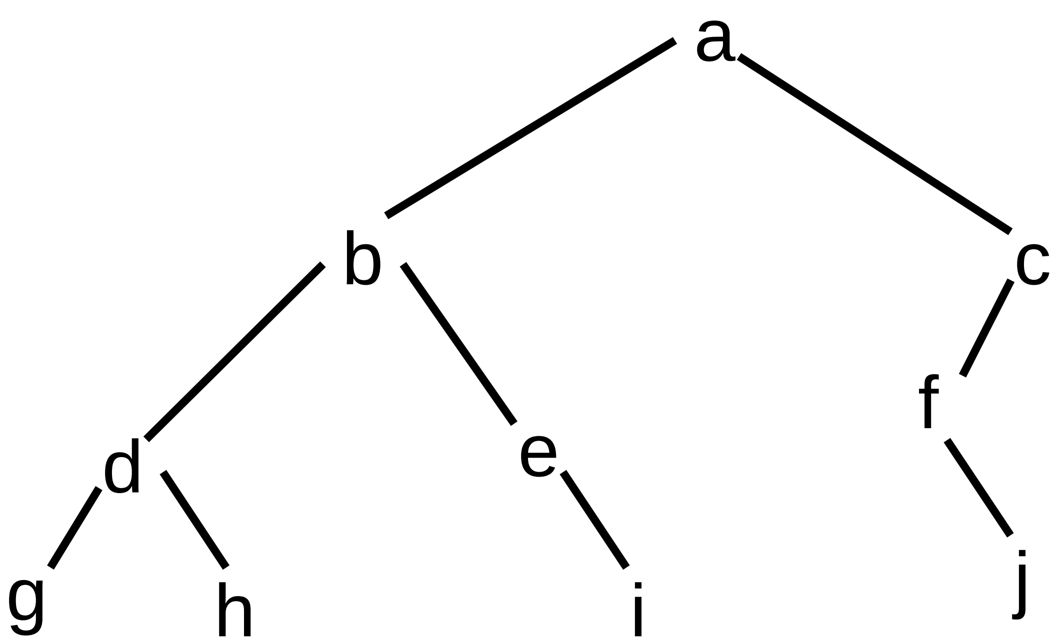
b c a

Postorder Example (visit = print)



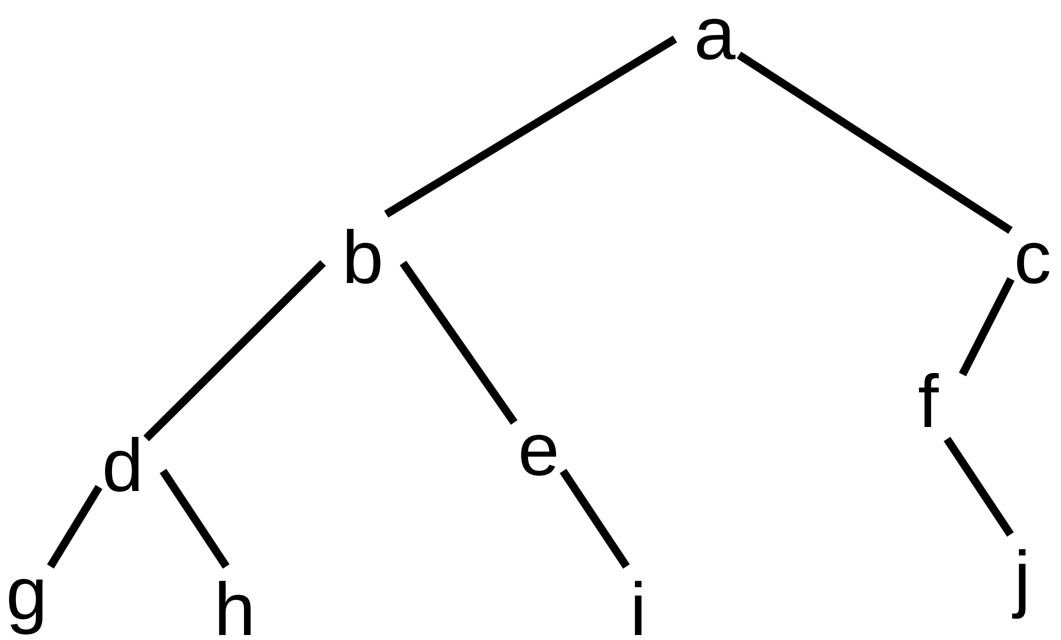
g h d i e b j f c a

Traversal Applications



- Make a clone.
- Determine height.
- Determine number of nodes.

Level-Order Example



a b c d e f g h i j

Binary Tree Construction

- Suppose that the elements in a binary tree are distinct.
- Can you construct the binary tree from which a given traversal sequence came?
- When a traversal sequence has more than one element, the binary tree is not uniquely defined.
- Therefore, the tree from which the sequence was obtained cannot be reconstructed uniquely.

Binary Search Trees

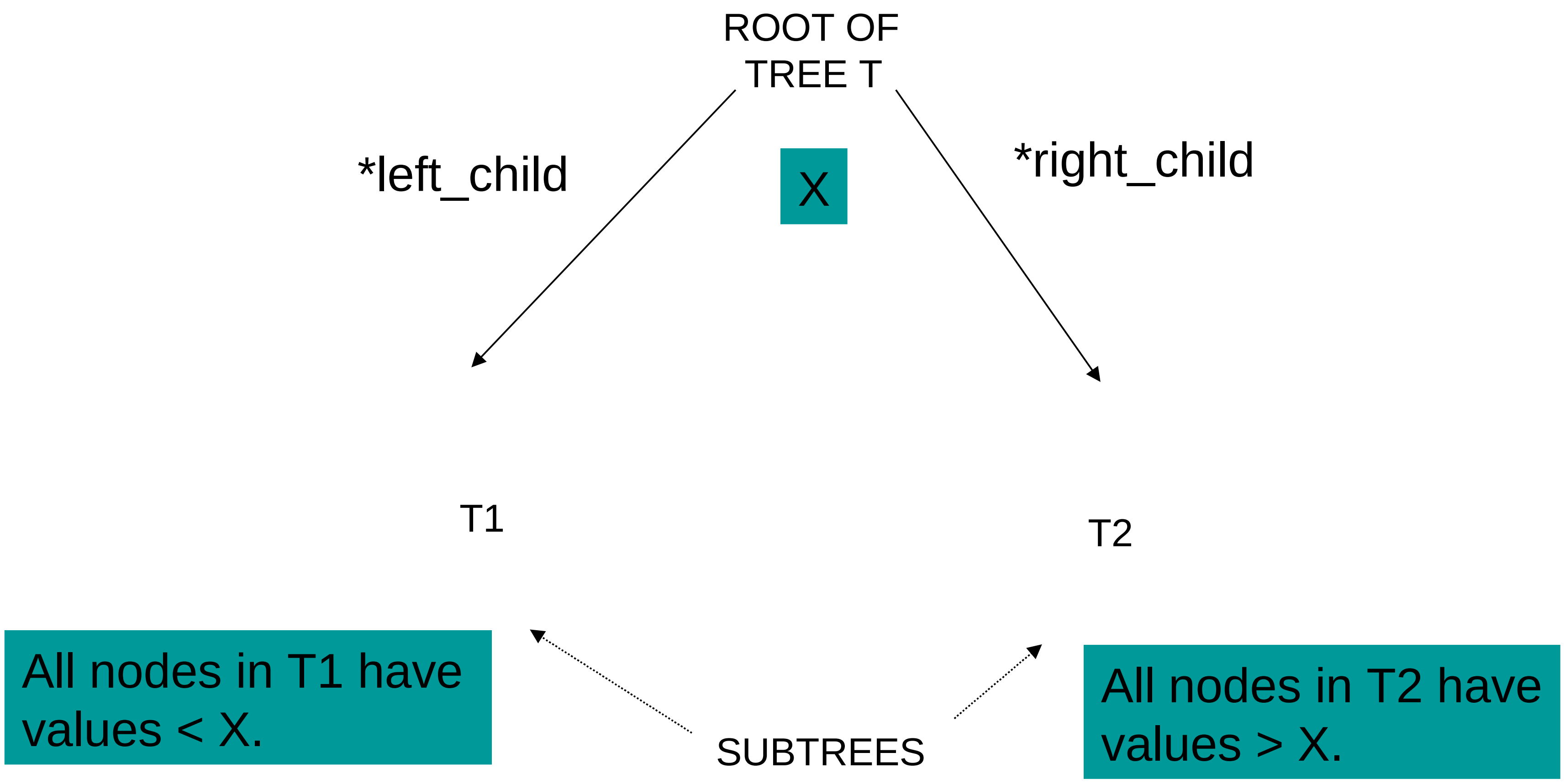
- Dictionary Operations:
  - get(key)
  - put(key, value)
  - remove(key)

Complexity Of Dictionary Operations

Data Structure	Worst Case	Expected
Hash Table	O(n)	O(1)
Binary Search Tree	O(n)	O(log n)
Balanced Binary Search Tree	O(log n)	O(log n)

n is number of elements in dictionary

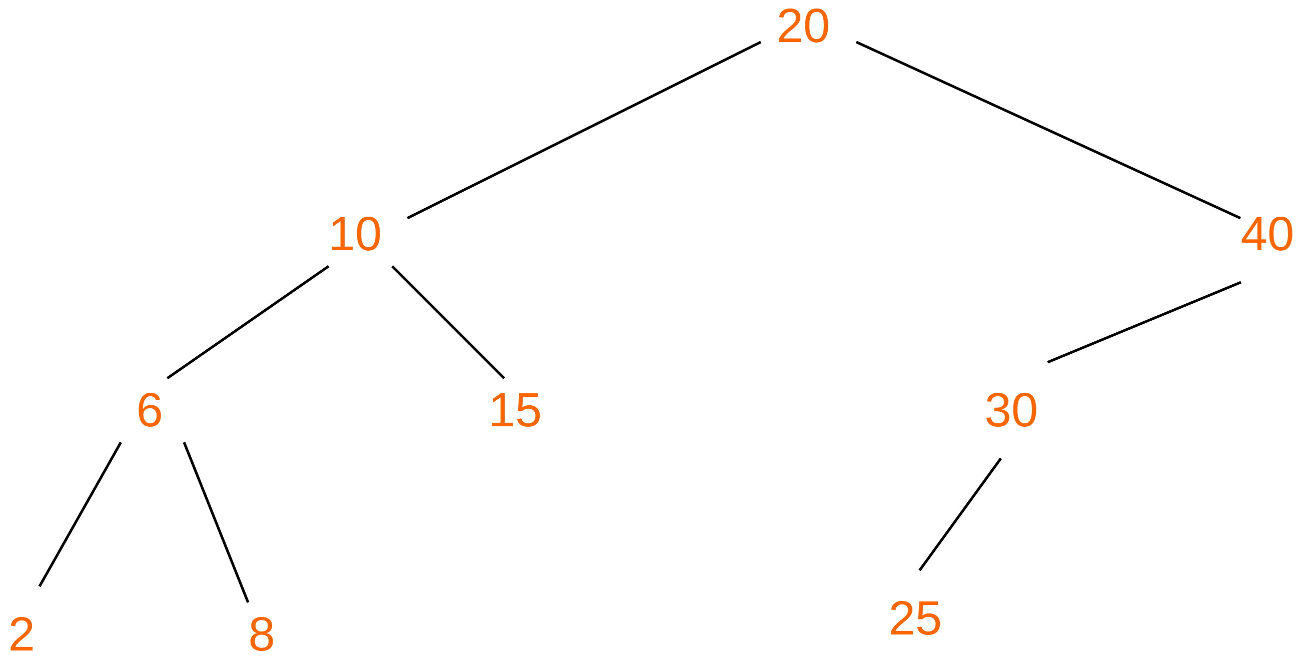
Binary Search Tree



Definition Of Binary Search Tree

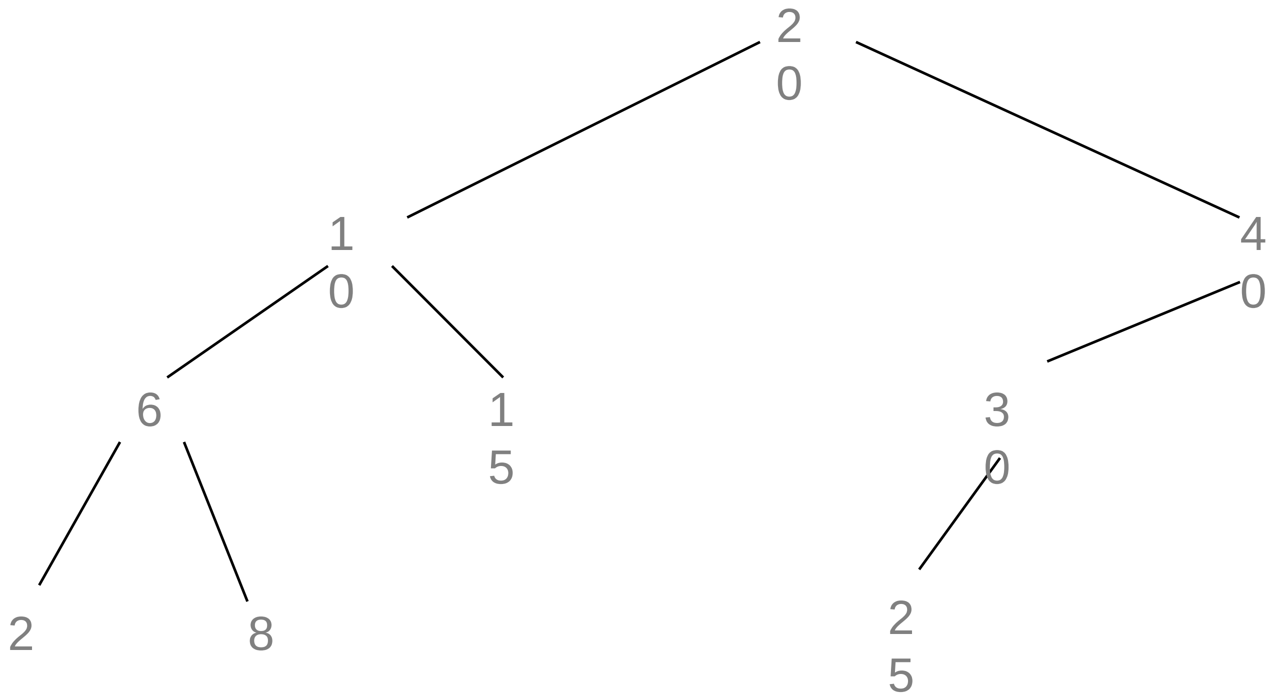
- A binary tree.
- Each node has a (key, value) pair.
- For every node x, all keys in the left subtree of x are smaller than that in x.
- For every node x, all keys in the right subtree of x are greater than that in x.

Example Binary Search Tree



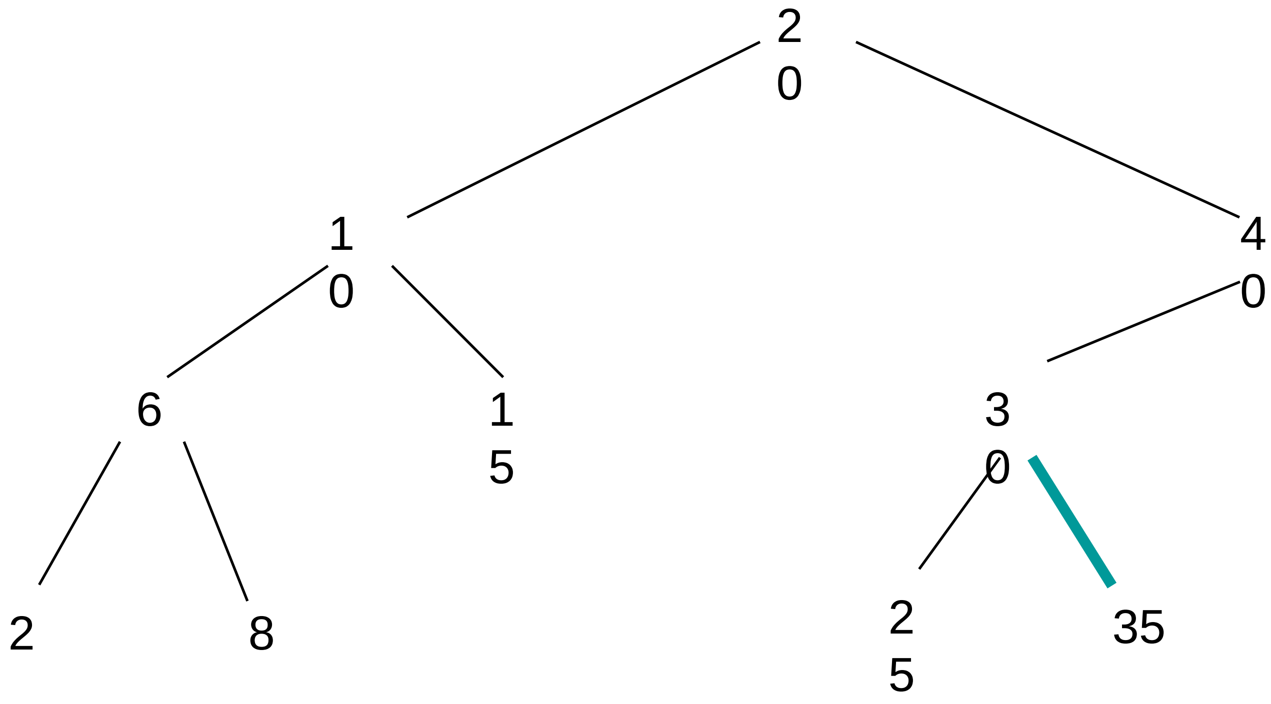
Only keys are shown.

The Operation get()



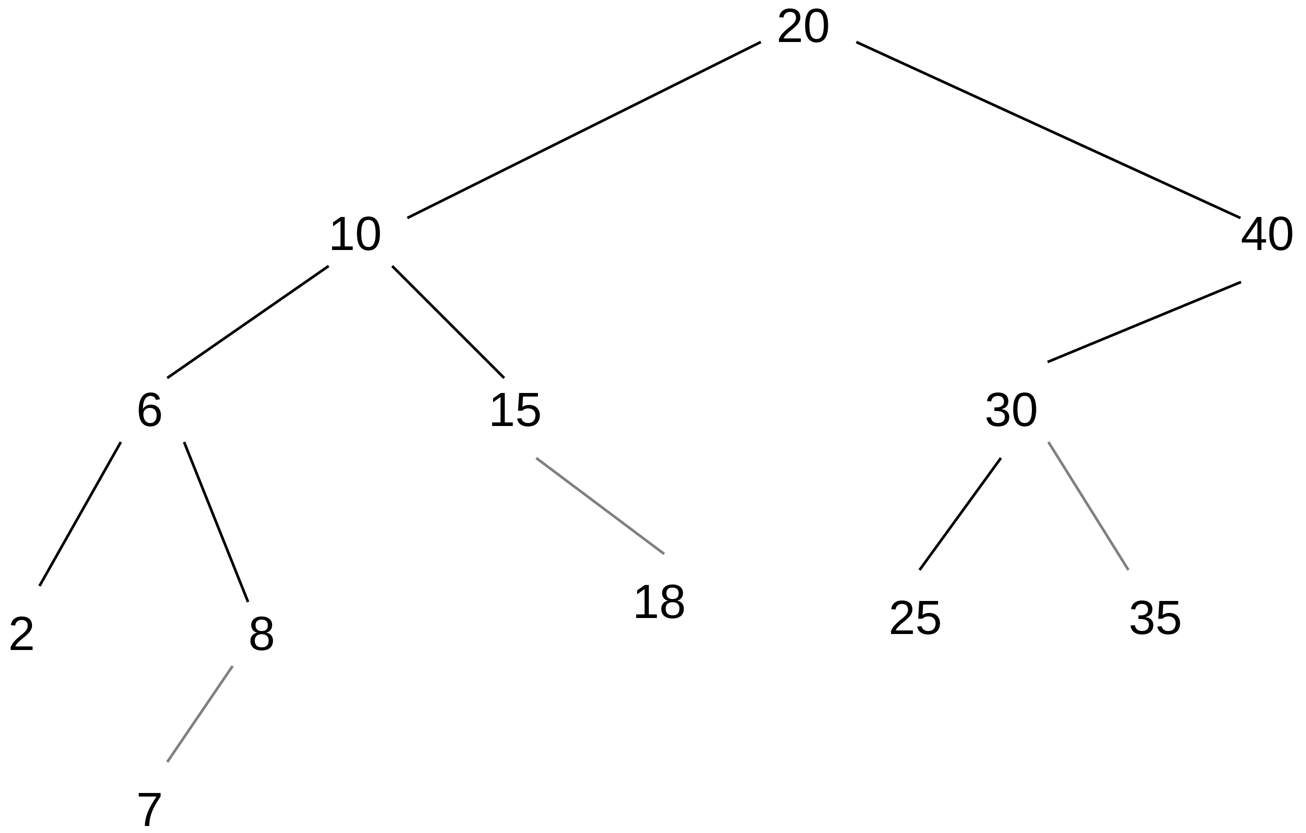
Complexity is O(height) = O(n), where n is number of nodes/elements.

The Operation put()



Put a pair whose key is 35.

The Operation put()



Complexity of put() is O(height).

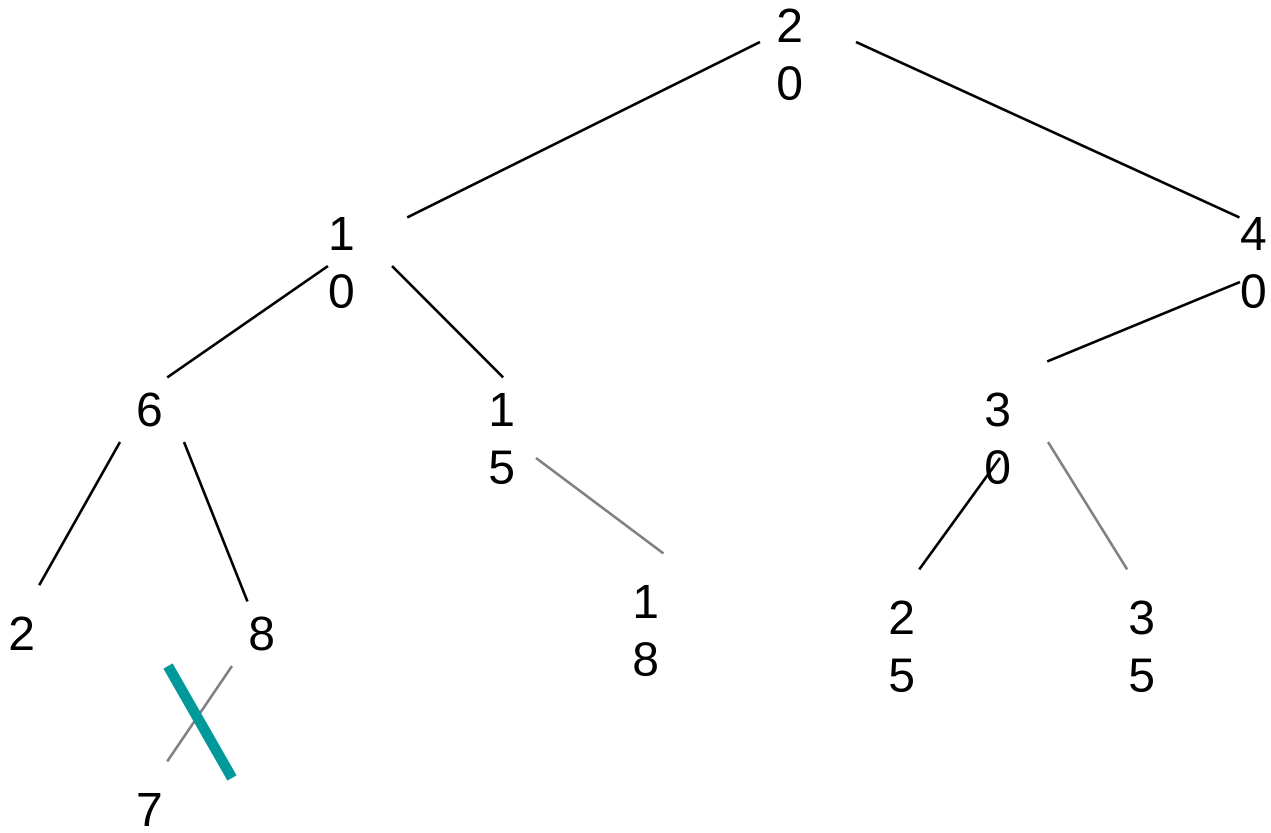
CADSL

The Operation remove()

Three cases:

- Element is in a leaf.
- Element is in a degree 1 node.
- Element is in a degree 2 node.

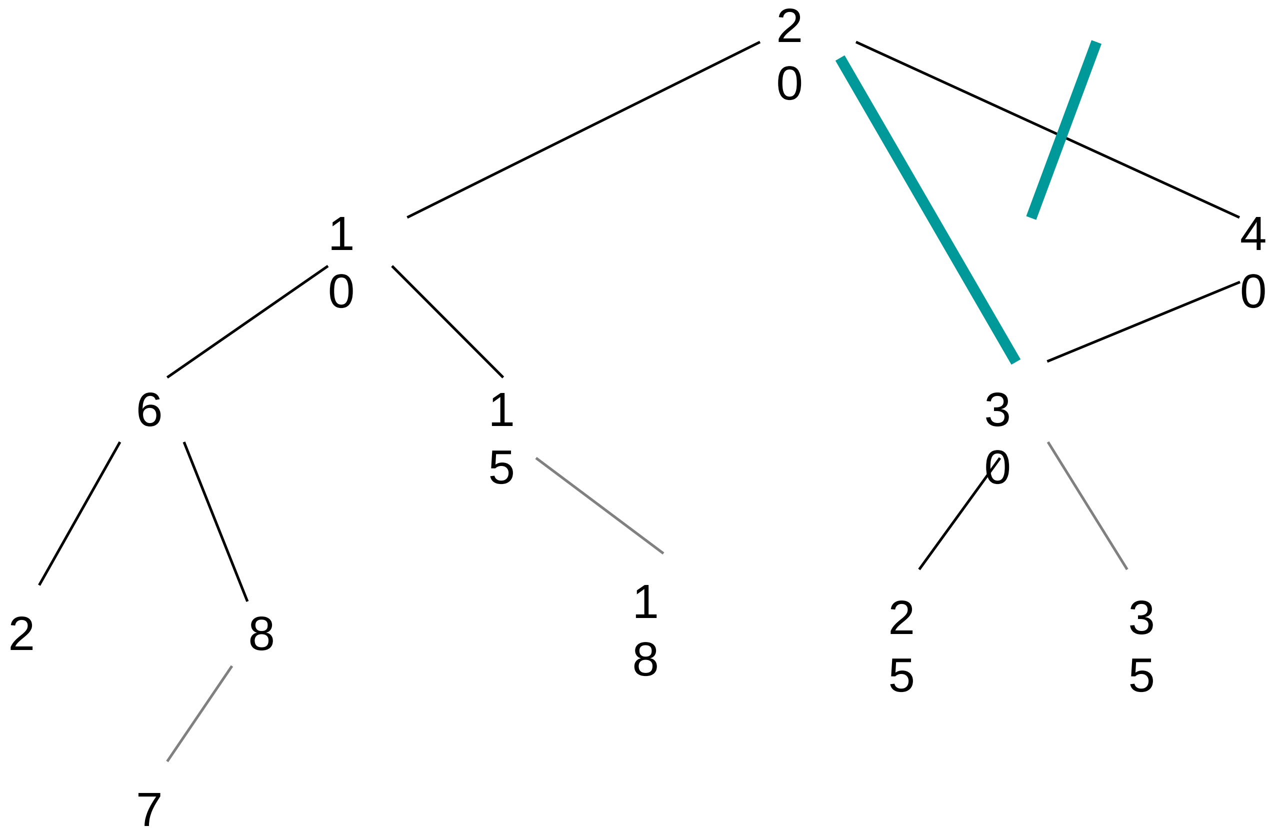
Remove From A Leaf



Remove a leaf element. key = 7

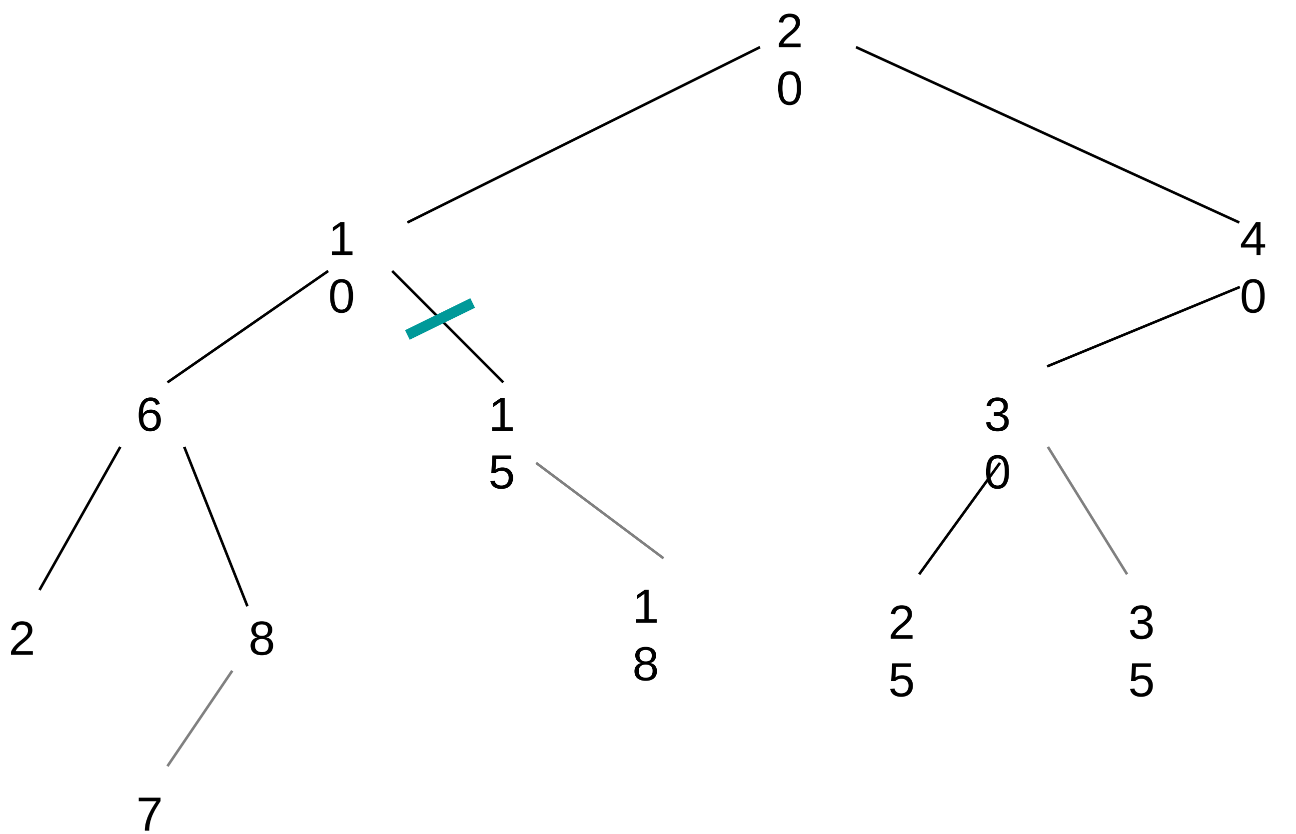
CADSL

Remove From A Degree 1 Node



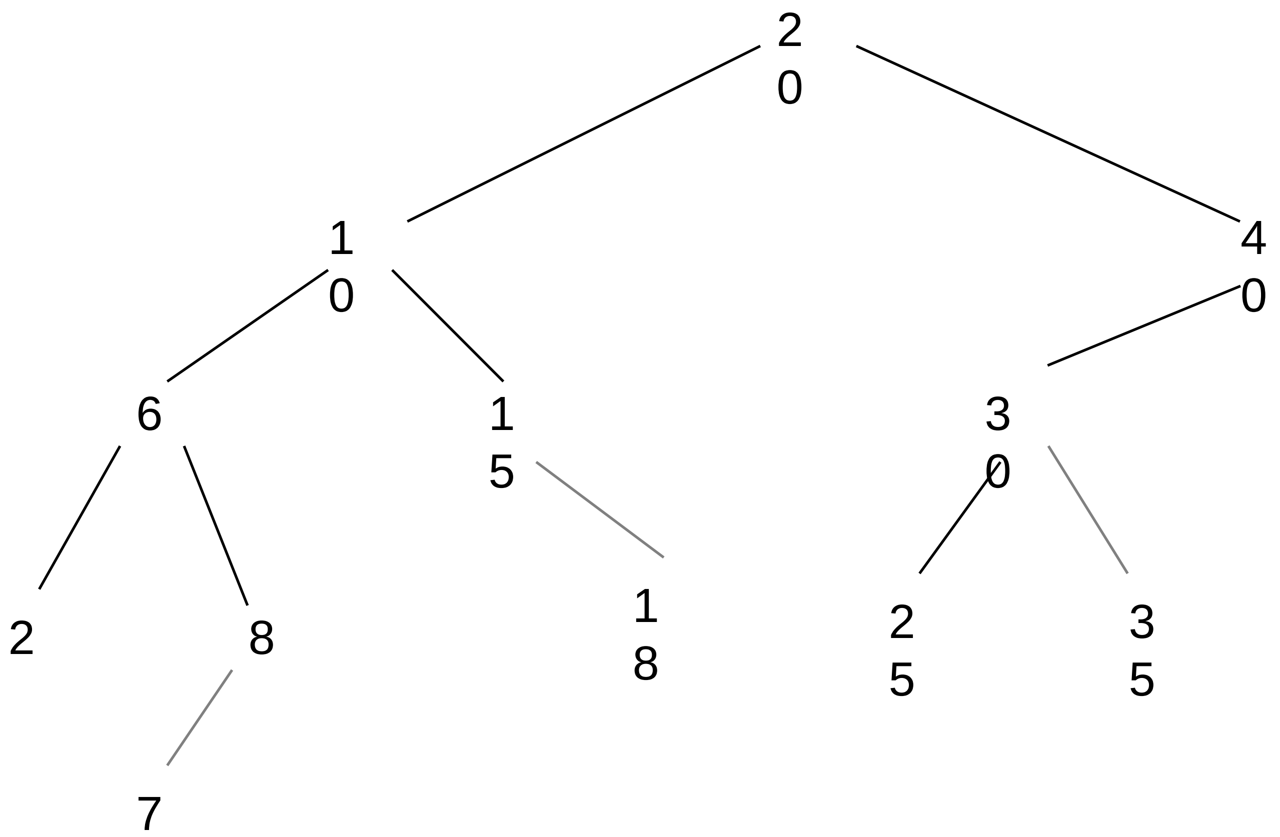
Remove from a degree 1 node. key = 40

Remove From A Degree 1 Node



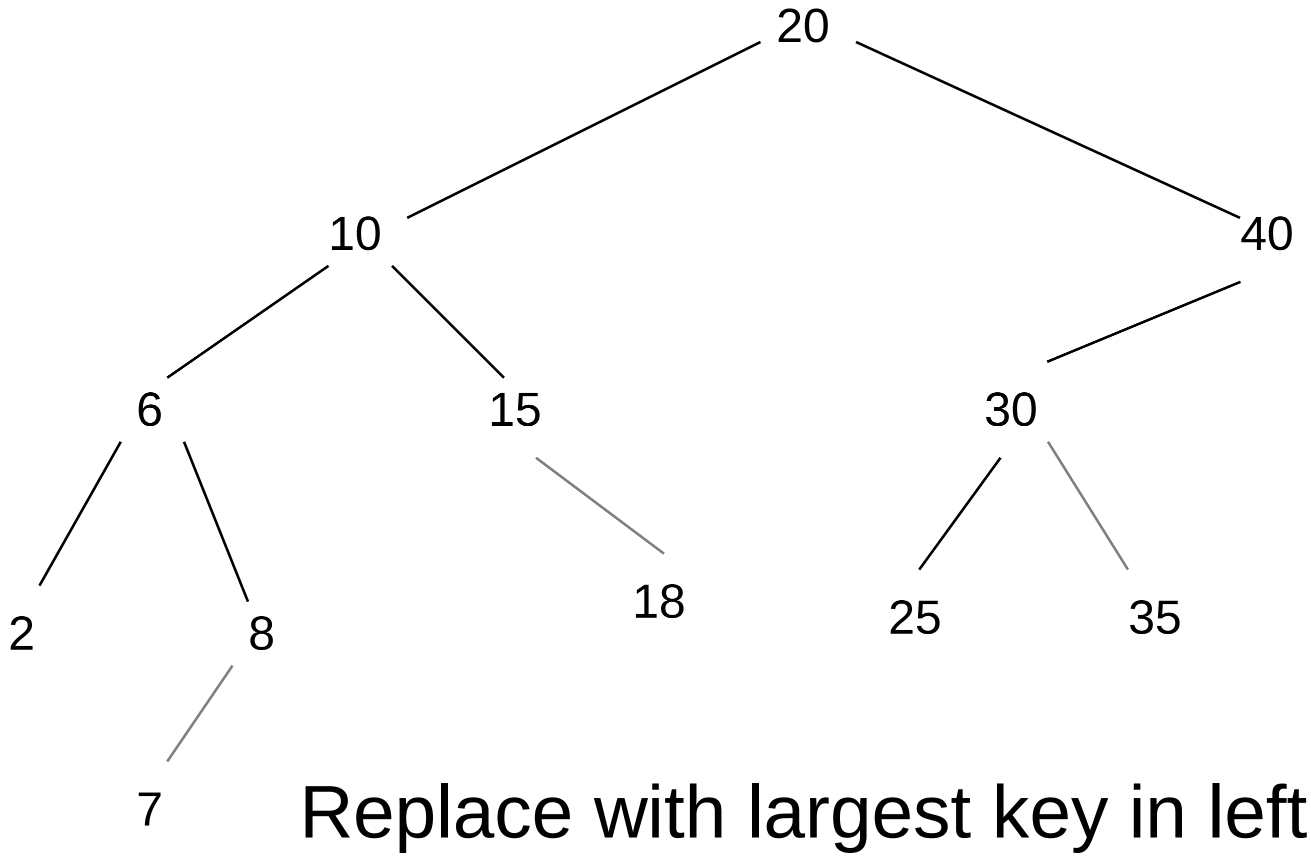
Remove from a degree 1 node. key = 15

Remove From A Degree 2 Node



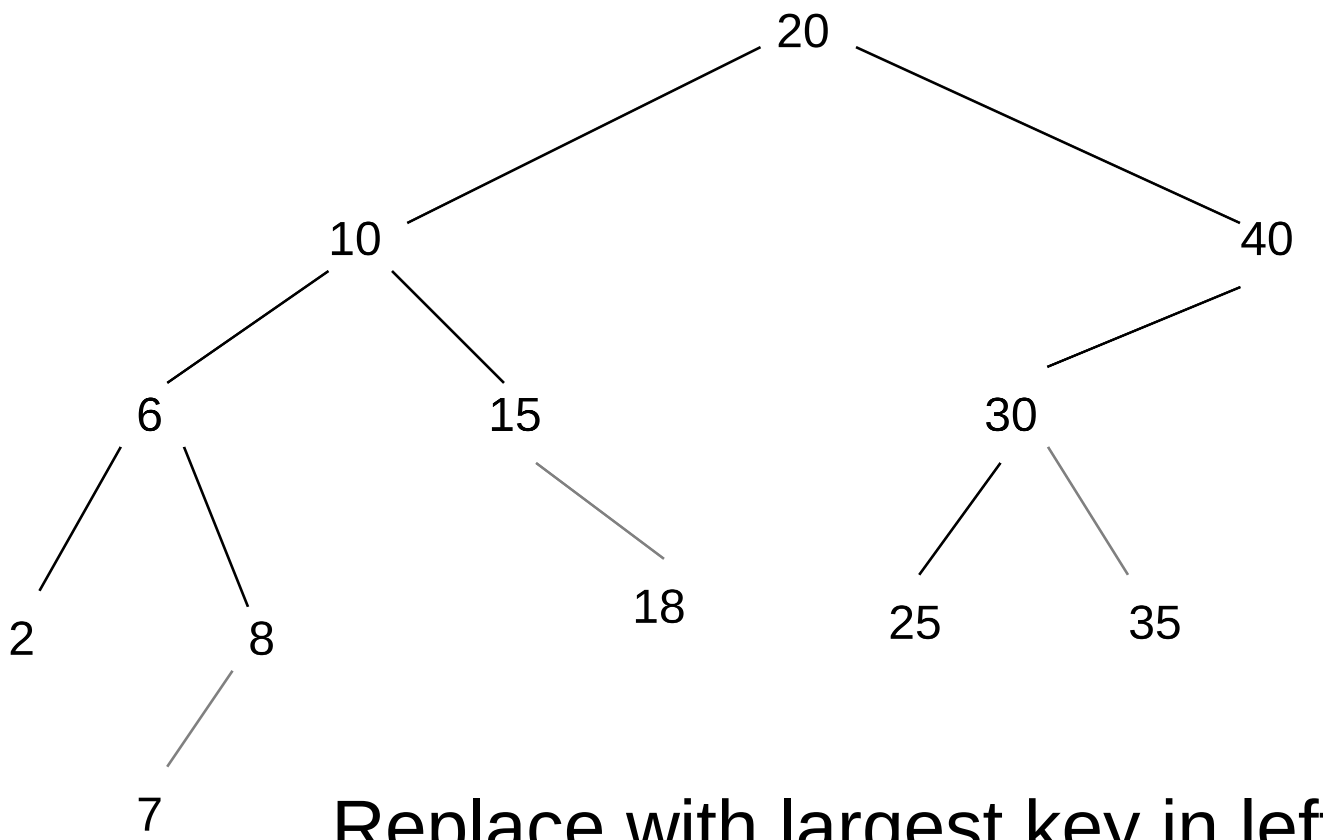
Remove from a degree 2 node. key = 10

Remove From A Degree 2 Node



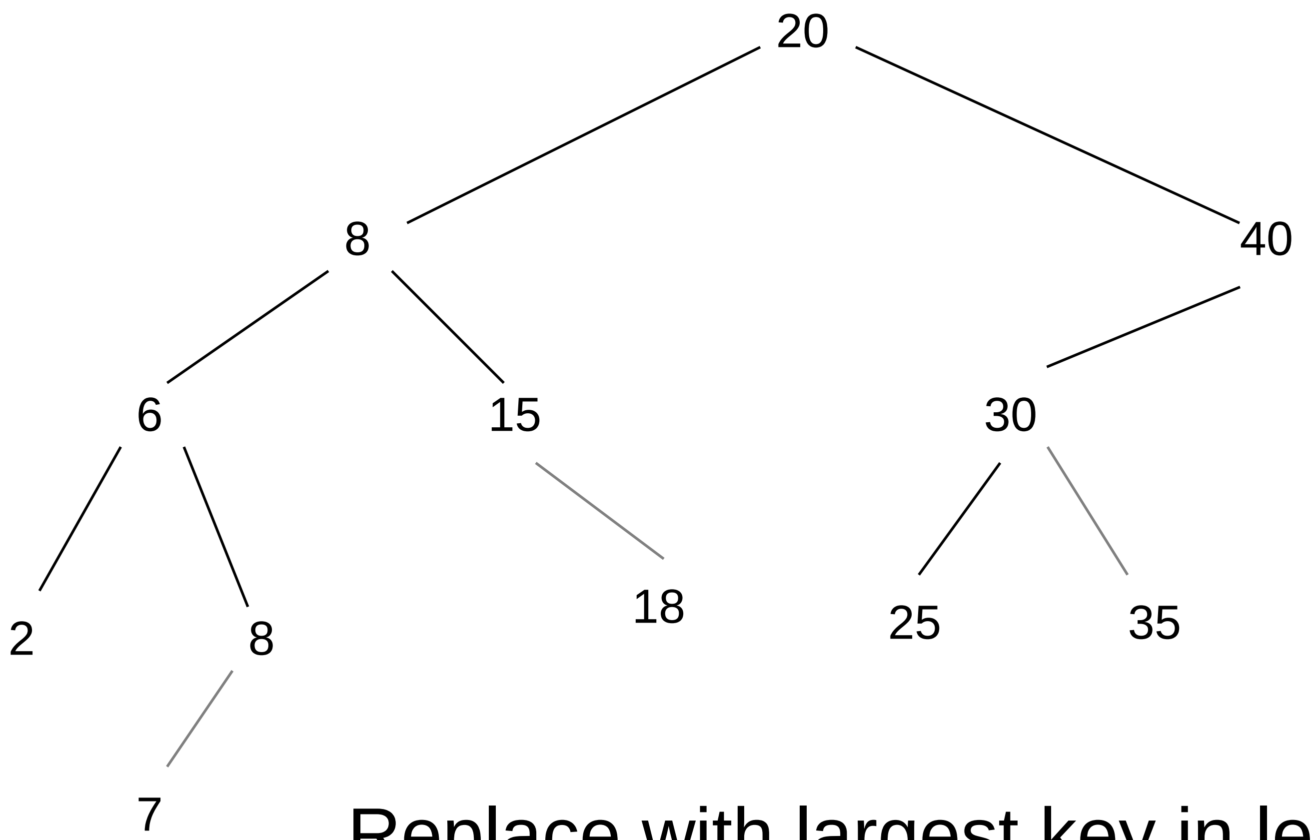
Replace with largest key in left subtree (or smallest in right subtree).

Remove From A Degree 2 Node



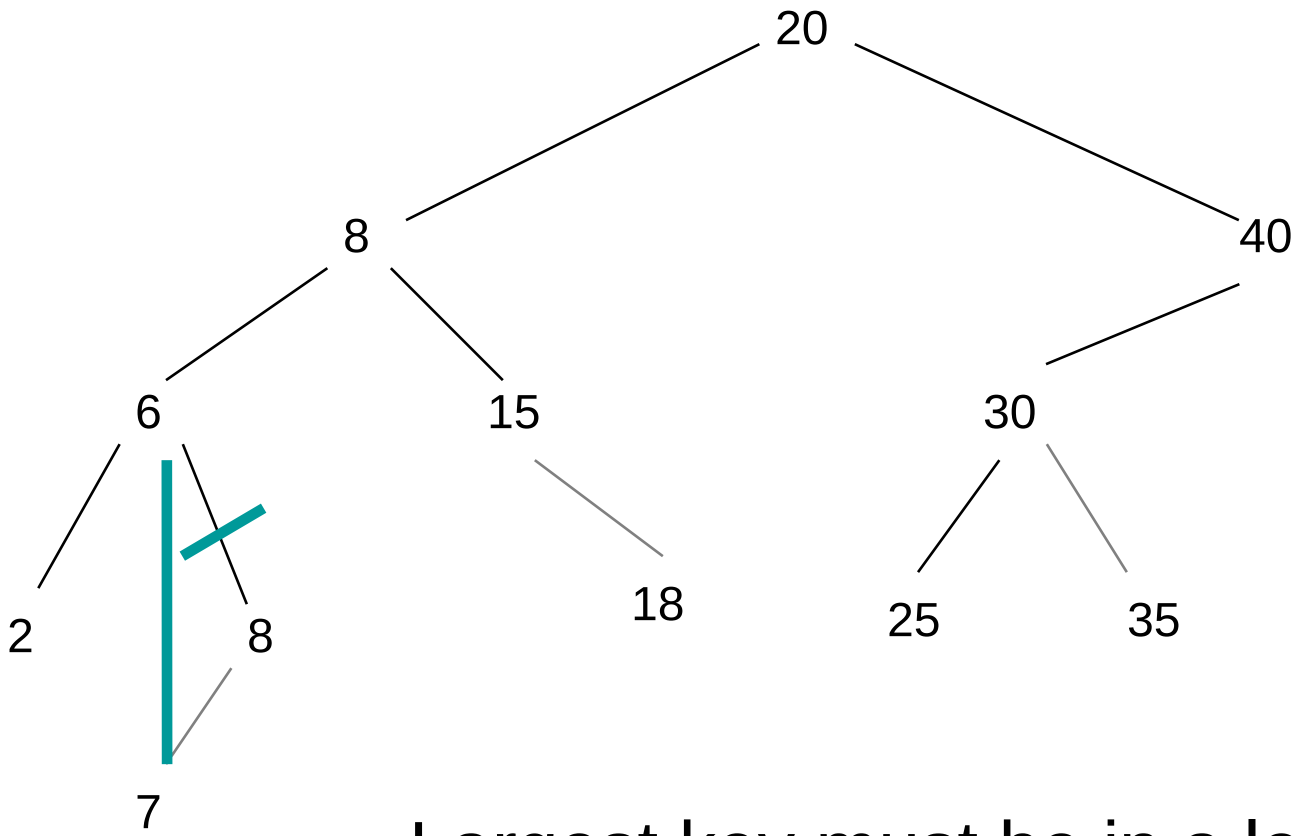
Replace with largest key in left subtree (or smallest in right subtree).

Remove From A Degree 2 Node



Replace with largest key in left subtree
(or smallest in right subtree).

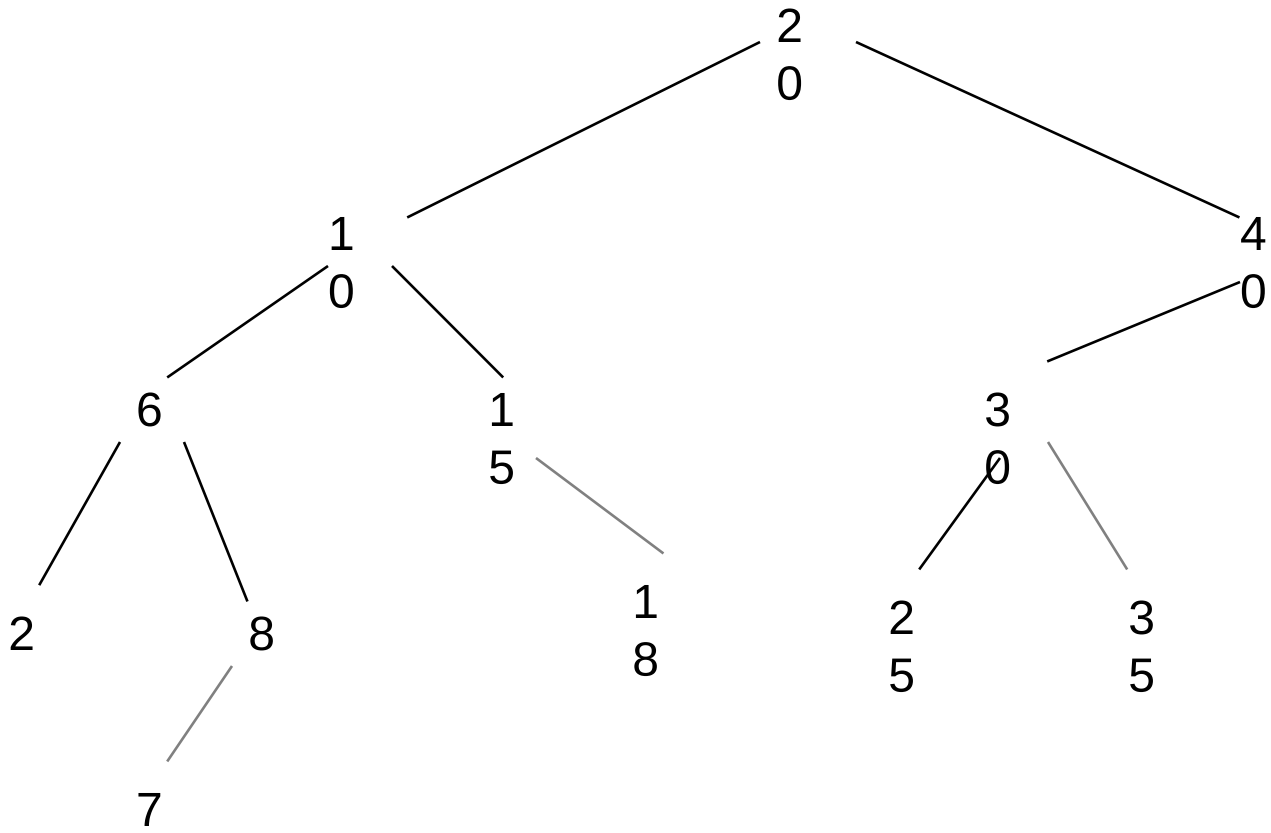
Remove From A Degree 2 Node



Largest key must be in a leaf or degree 1 node.

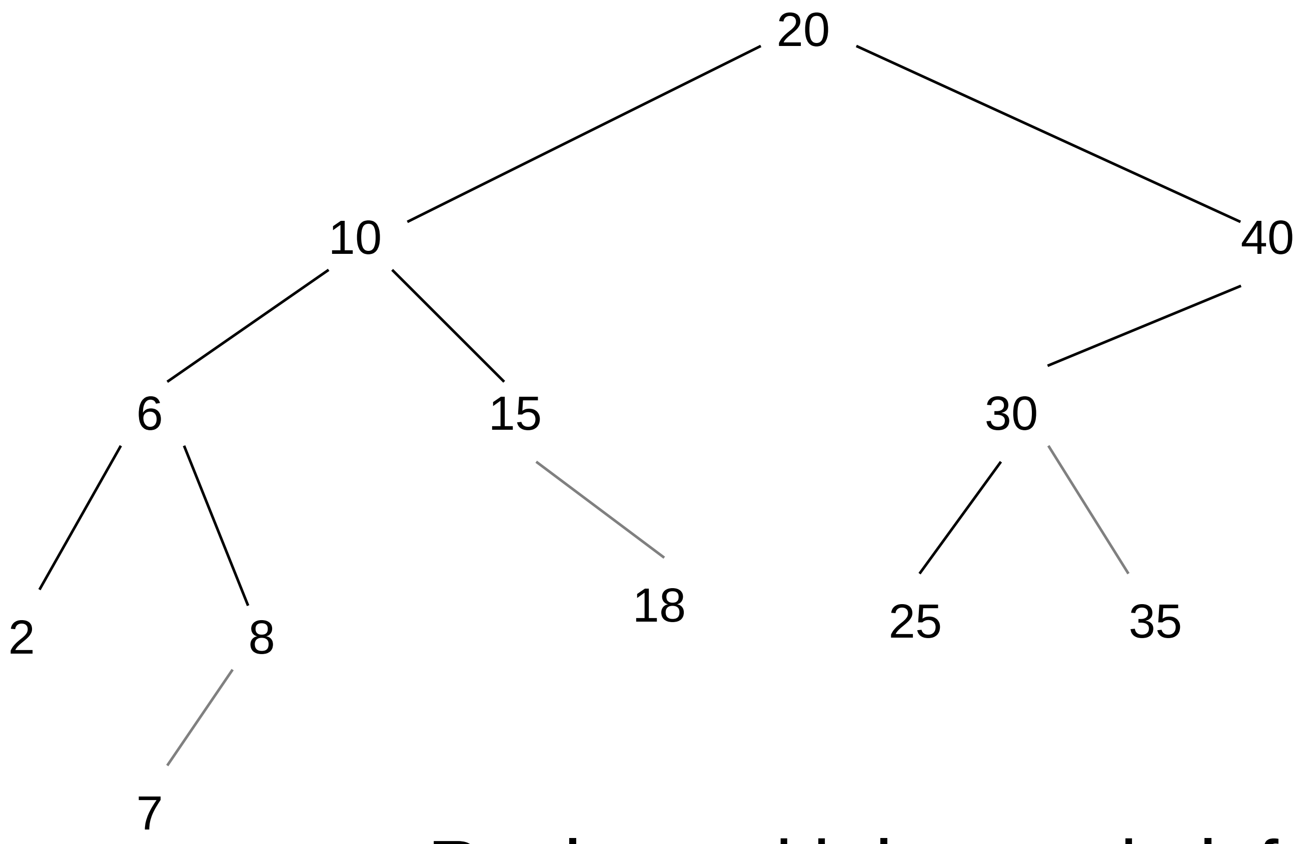
CADSL

Another Remove From A Degree 2 Node



Remove from a degree 2 node. key = 20

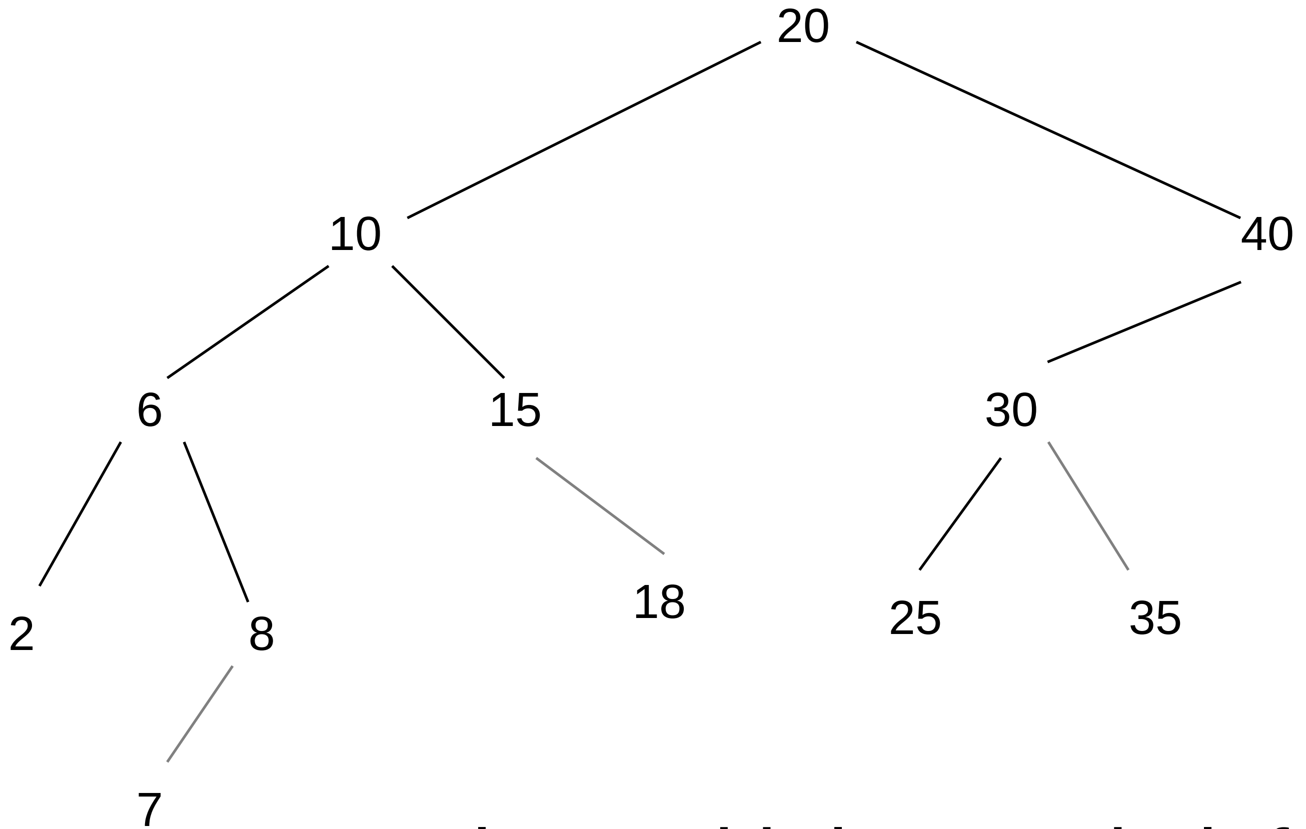
Remove From A Degree 2 Node



Replace with largest in left subtree.

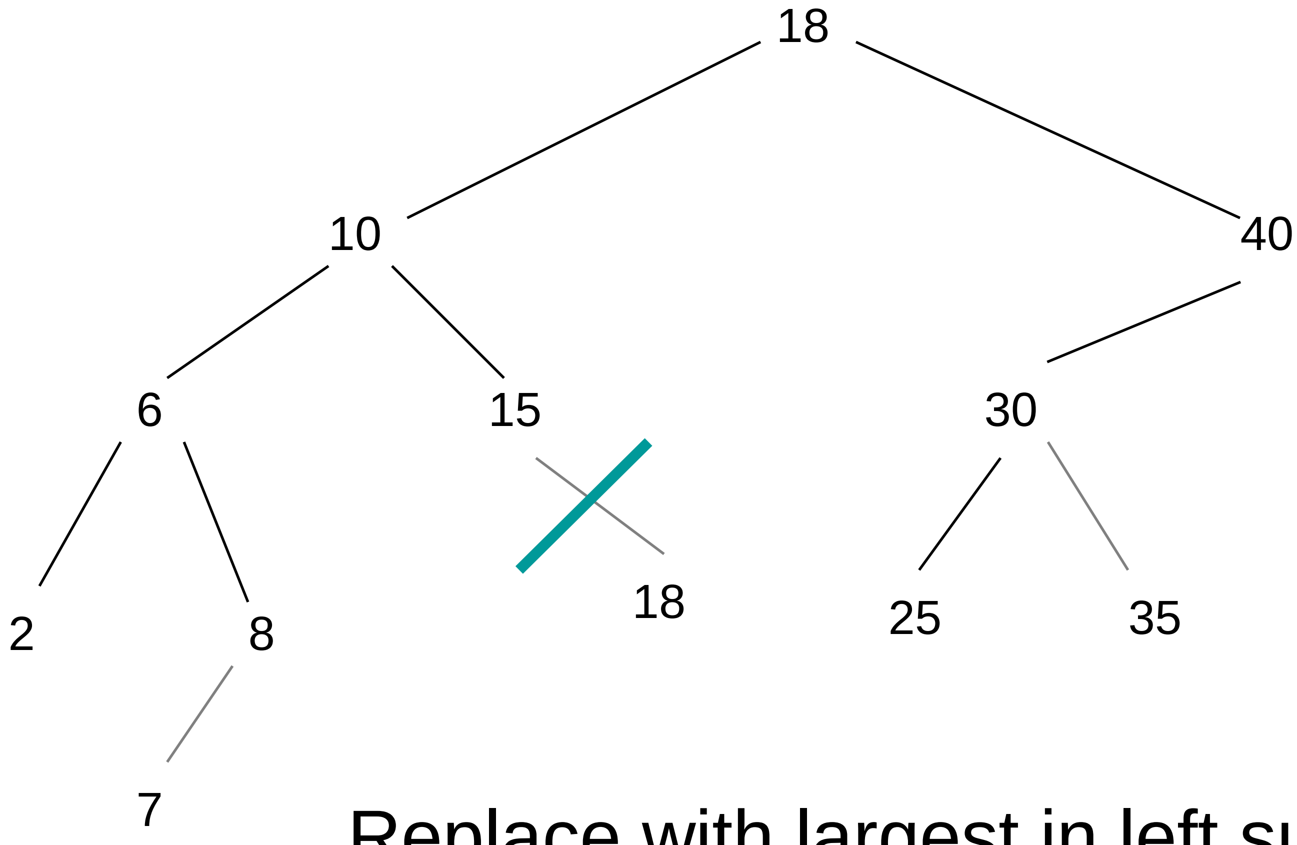
CADSL

Remove From A Degree 2 Node



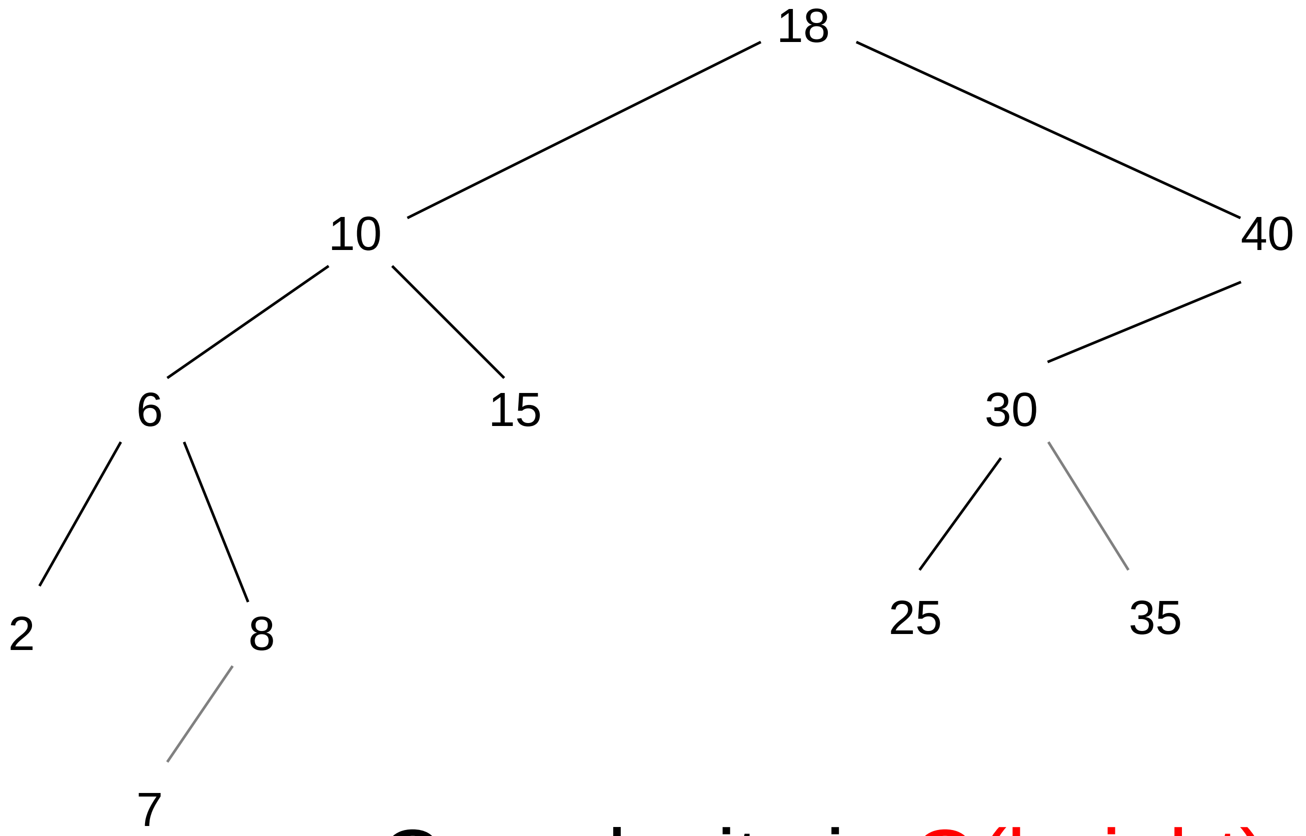
Replace with largest in left subtree.

Remove From A Degree 2 Node



Replace with largest in left subtree.

Remove From A Degree 2 Node



Complexity is O(height).

Analysis

- The running time of these operations is *O(d)*, where *d* is the depth of the node containing the accessed item.
- What is the average depth of the nodes in a binary search tree? It depends on how well balanced the tree is.