DS286 | 2016-09-28,30

L15,16: Binary Tree Construction & Search Trees

Yogesh Simmhan
simmhan@cds.iisc.ac.in

Slides courtesy Venkatesh Babu, CDS & Sahni textbook

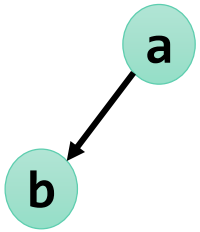
Tree Construction

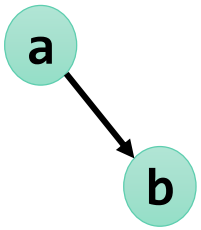
Binary Tree Construction

- *Can you construct the binary tree based on a given traversal sequence?*
- Assume the elements in a binary tree are distinct
- 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.

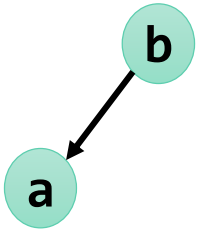
Examples

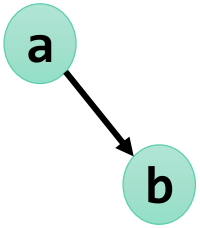
preorder = ab



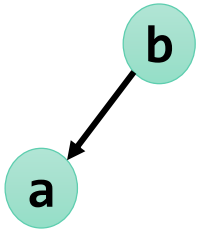


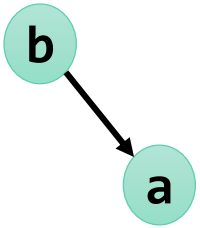
inorder = ab



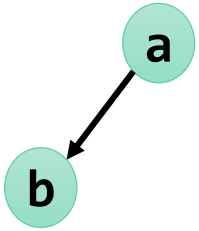


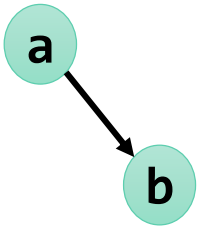
postorder = ab





level order = ab





Binary Tree Construction

- *Can you construct the binary tree, given two traversal sequences?*
- Depends on which two sequences are given

Preorder And Postorder

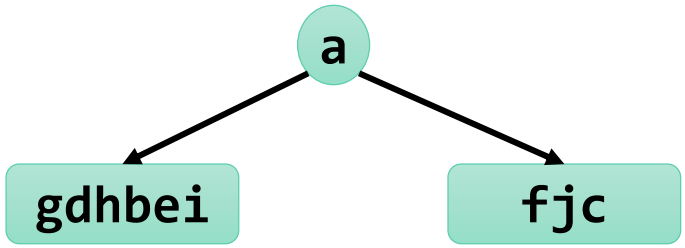
preorder = ab

postorder = ba

- Preorder and postorder do not uniquely define a binary tree.
- Nor do preorder and level order (same example)
- Nor do postorder and level order (same example)

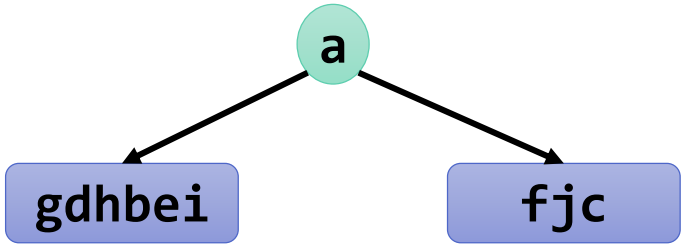
Inorder And Preorder

- inorder = g d h b e i **a** f j c
- preorder = **a** b d g h e i c f j
- Scan the preorder left to right using the inorder to separate left and right subtrees.
- a is the root of the tree; gdhbei are in the left subtree; fjc are in the right subtree

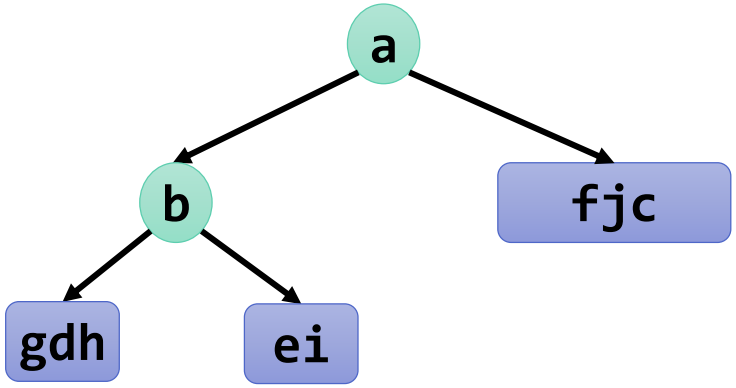


Inorder And Preorder

- inorder = g d h b e i **a** f j c
- preorder = **a** b d g h e i c f j

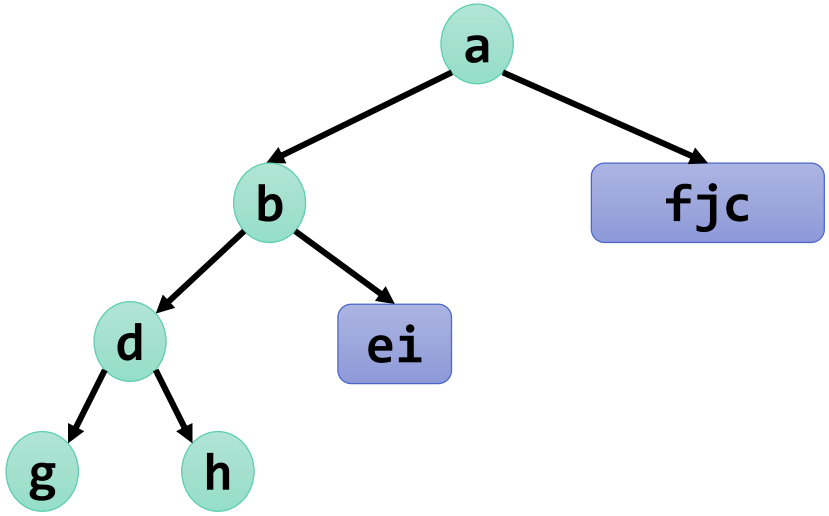


- inorder = g d h **b** e i a f j c
- preorder = a **b** d g h e i c f j

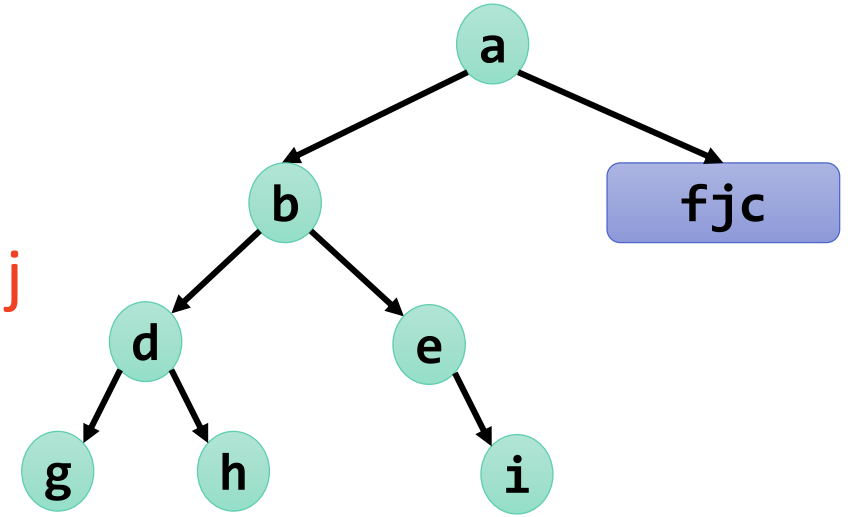


Inorder And Preorder

- inorder = g **d** h b e i a f j c
- preorder = a b **d** g h e i c f j

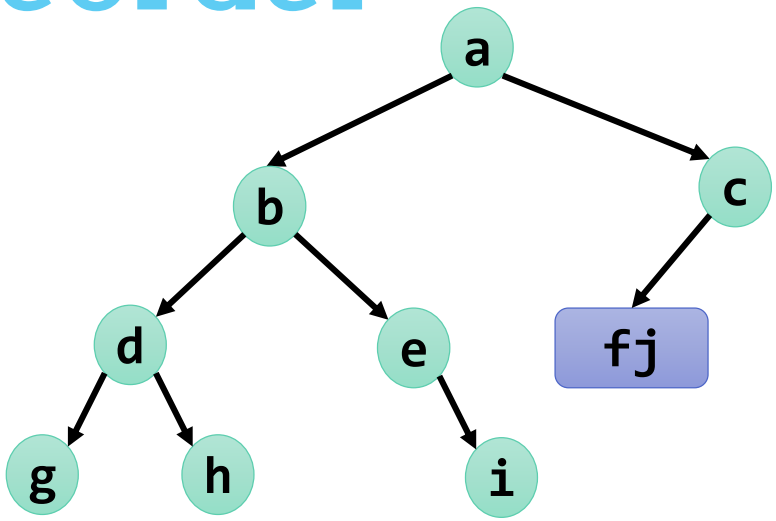


- inorder = g d h b **e** i a f j c
- preorder = a b d g h **e** i c f j

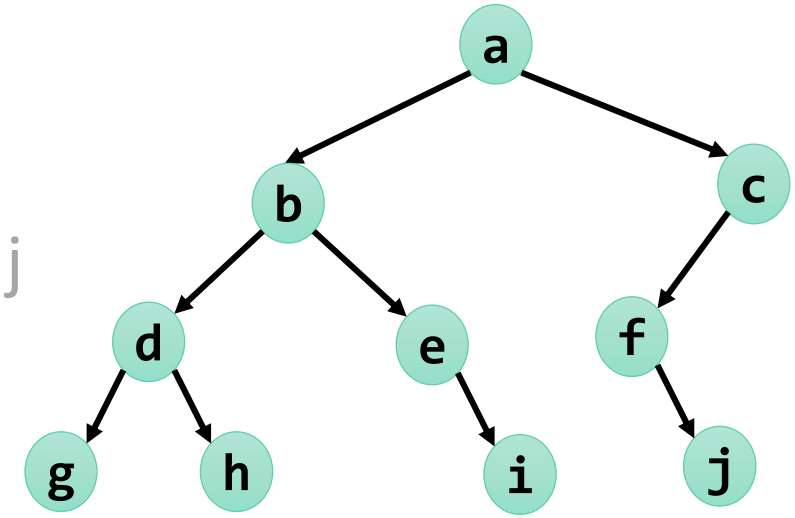


Inorder And Preorder

- inorder = g d h b e i a f j c
- preorder = a b d g h e i c f j



- inorder = g d h b e i a f j c
- preorder = a b d g h e i c f j



Inorder And Postorder

- Scan postorder from right to left using inorder to separate left and right subtrees
- inorder = g d h b e i **a** f j c
- postorder = g h d i e b j f c **a**
- Tree root is a; gdhbei are in left subtree; fjc are in right subtree.

Inorder And Level Order

- Scan level order from left to right using inorder to separate left and right subtrees.
- inorder = g d h b e i **a** f j c
- level order = **a** b c d e f g h i j
- Tree root is a; gdhbei are in left subtree; fjc are in right subtree.
- Next level roots are b and c; gdh,ei and fj,Φ are in their left,right subtrees

Binary Search Tree (BST)

Binary Search Trees

- Dictionary Operations
  - `find(key)`
  - `insert(key, value)`
  - `erase(key)`
- Additional Operations
  - `ascend()`

Complexity Of Dictionary Operations find(), insert(), erase()

- Given **n** elements in the dictionary

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)

Complexity Of Dictionary Operations delete(), ascend()

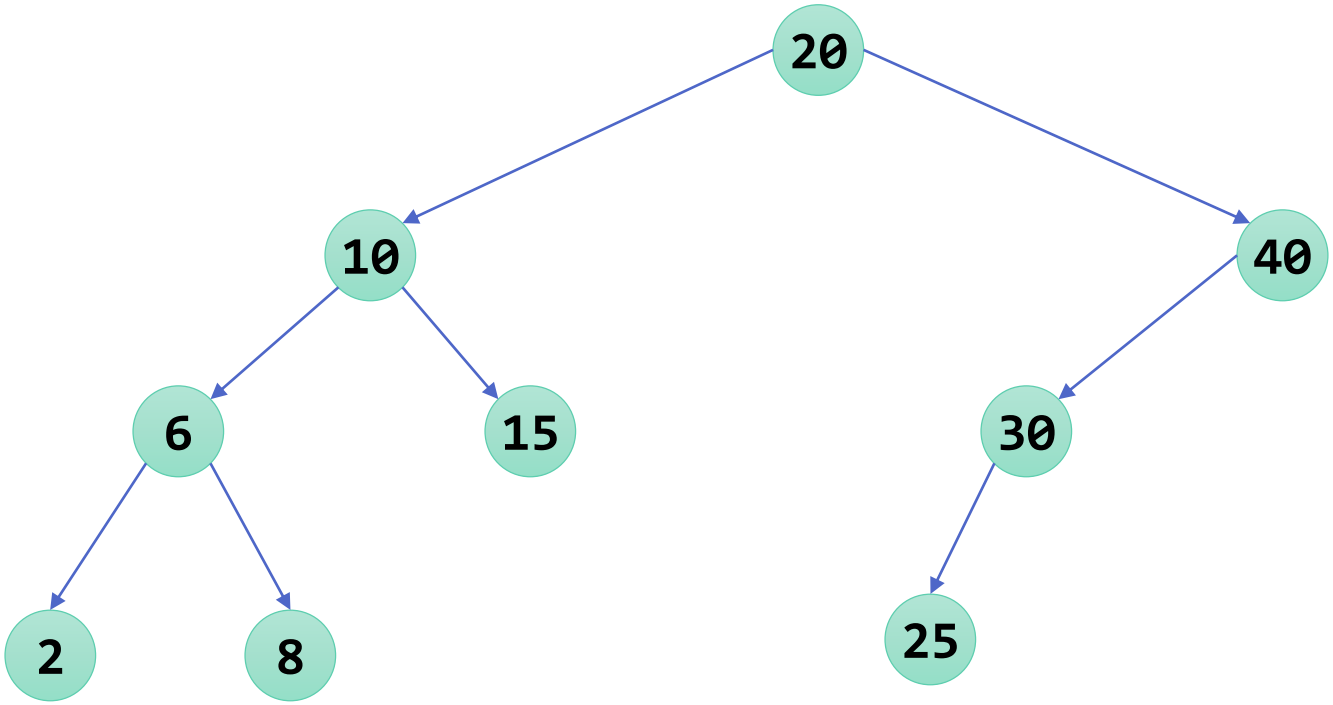
- Given **n** elements in the dictionary, **b** buckets

Data Structure	**Worst Case**	**Expected**
Hash Table	O(b+n.log n)	O(b+n.log n)
Binary Search Tree	O(n)	O(n)
Balanced Binary Search Tree	O(n)	O(log n)

Definition Of Binary Search Tree

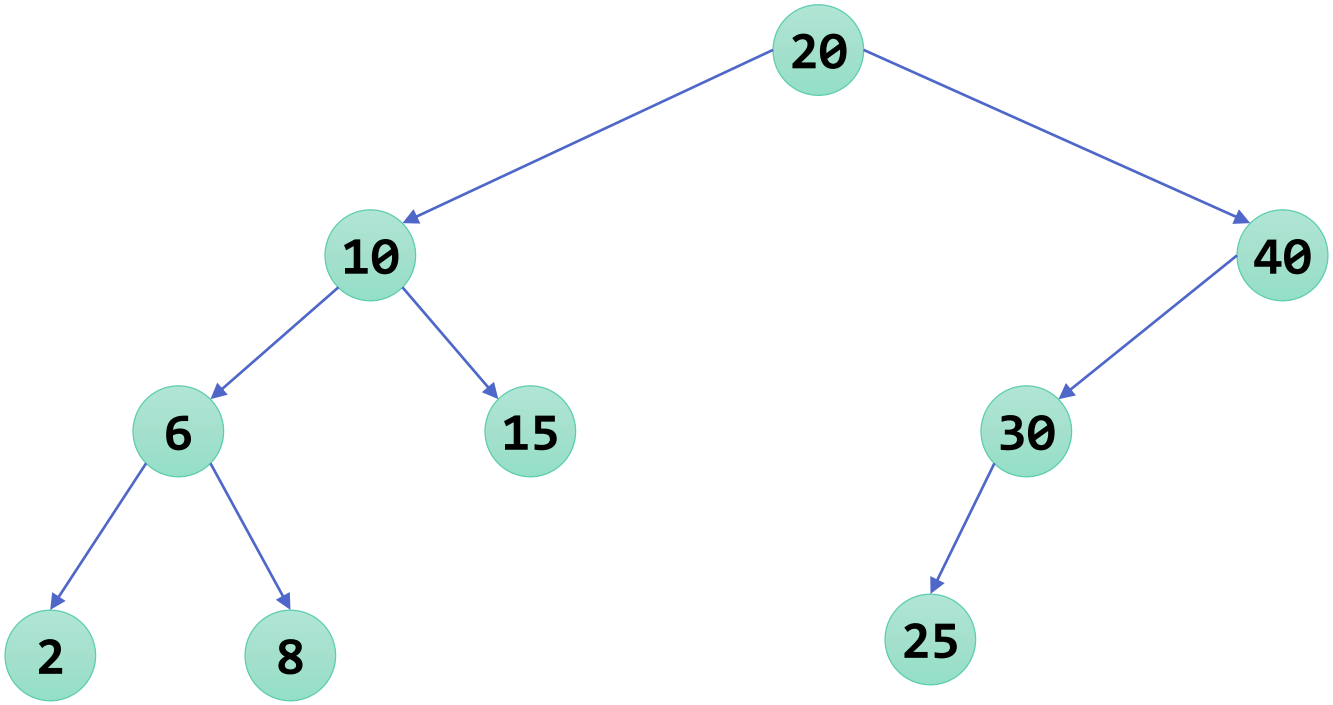
- A binary tree
- Each node has a **(key, value)** pair
- For every node x, all keys in the *left subtree* of x are *smaller* than that in x
- For every node x, all keys in the *right subtree* of x are *greater* than that in x

Example Binary Search Tree



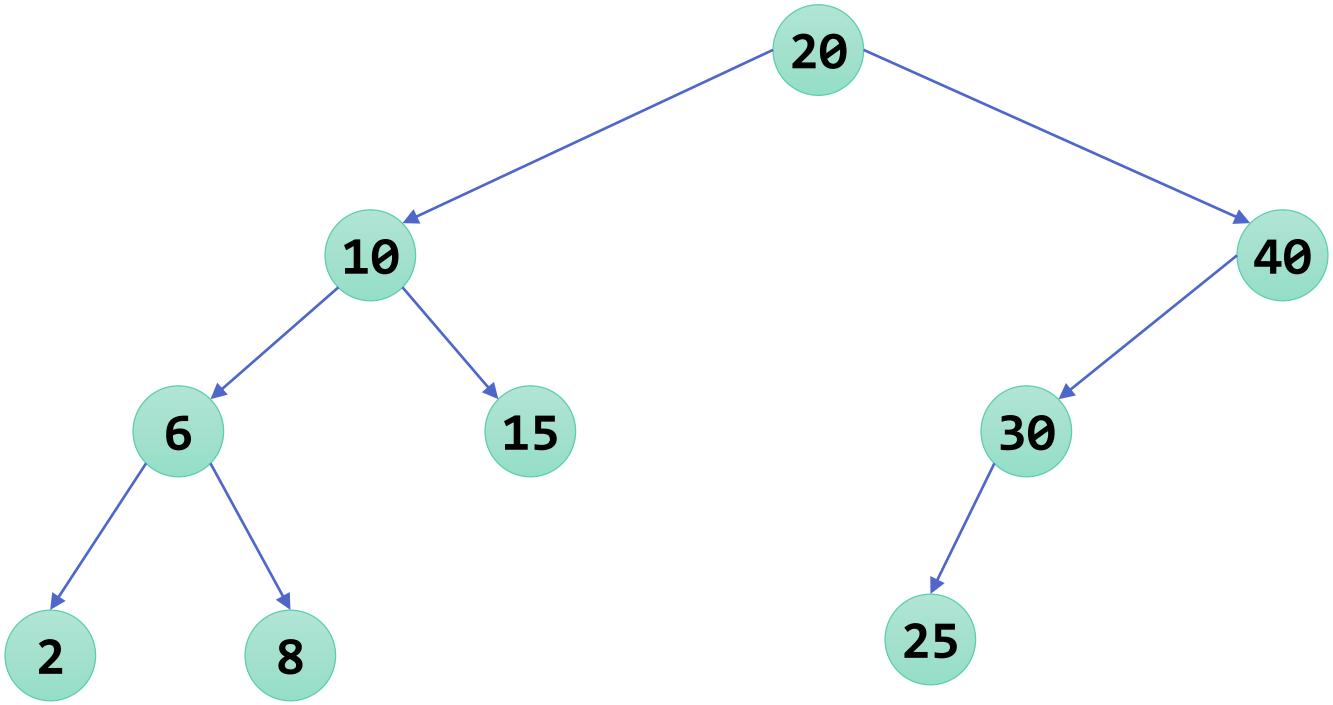
Only keys are shown.

The Operation find()



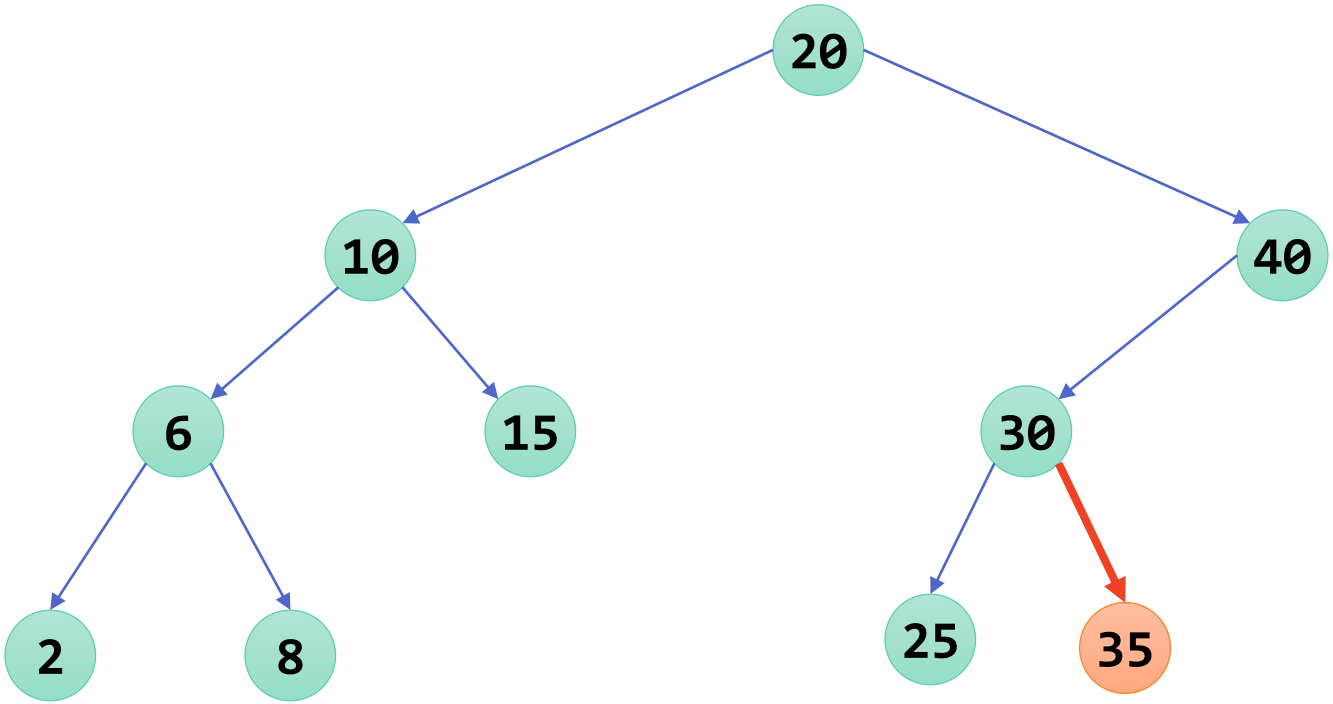
Complexity is O(height) = O(n), where n is the number of nodes/elements.

The Operation ascend()



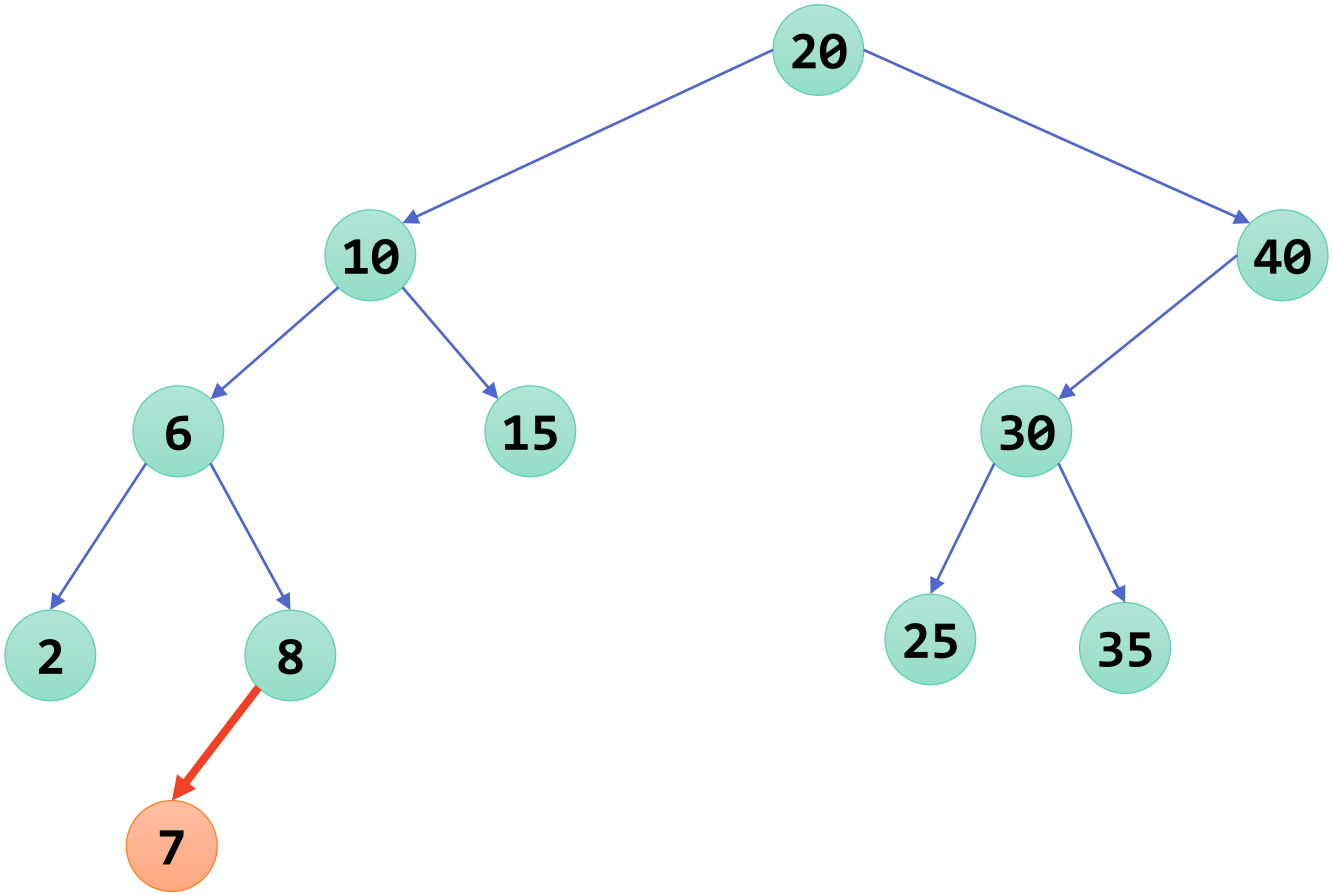
Do an inorder traversal. O(n) time.

The Operation insert()



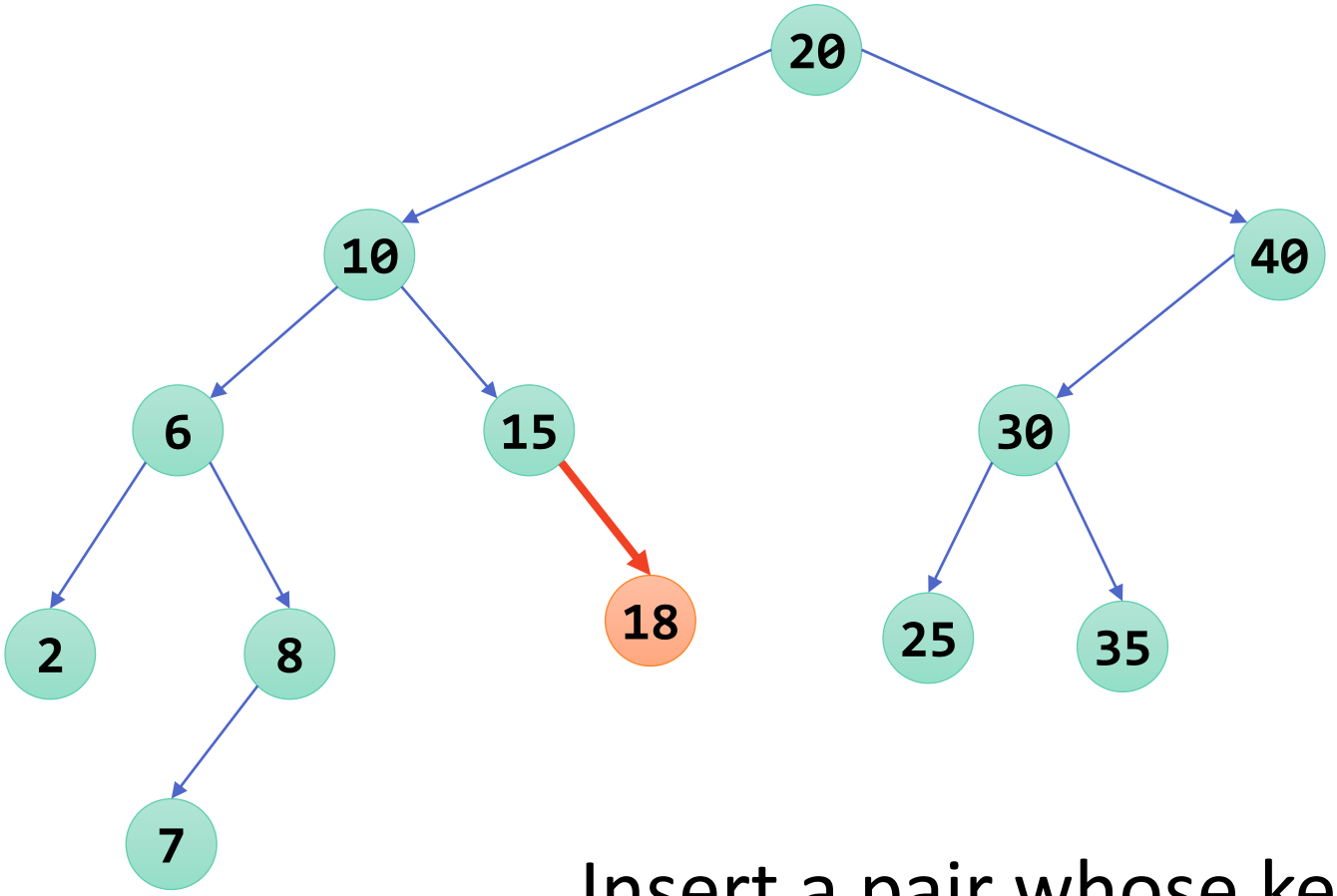
Insert a pair whose key is 35.

The Operation insert()



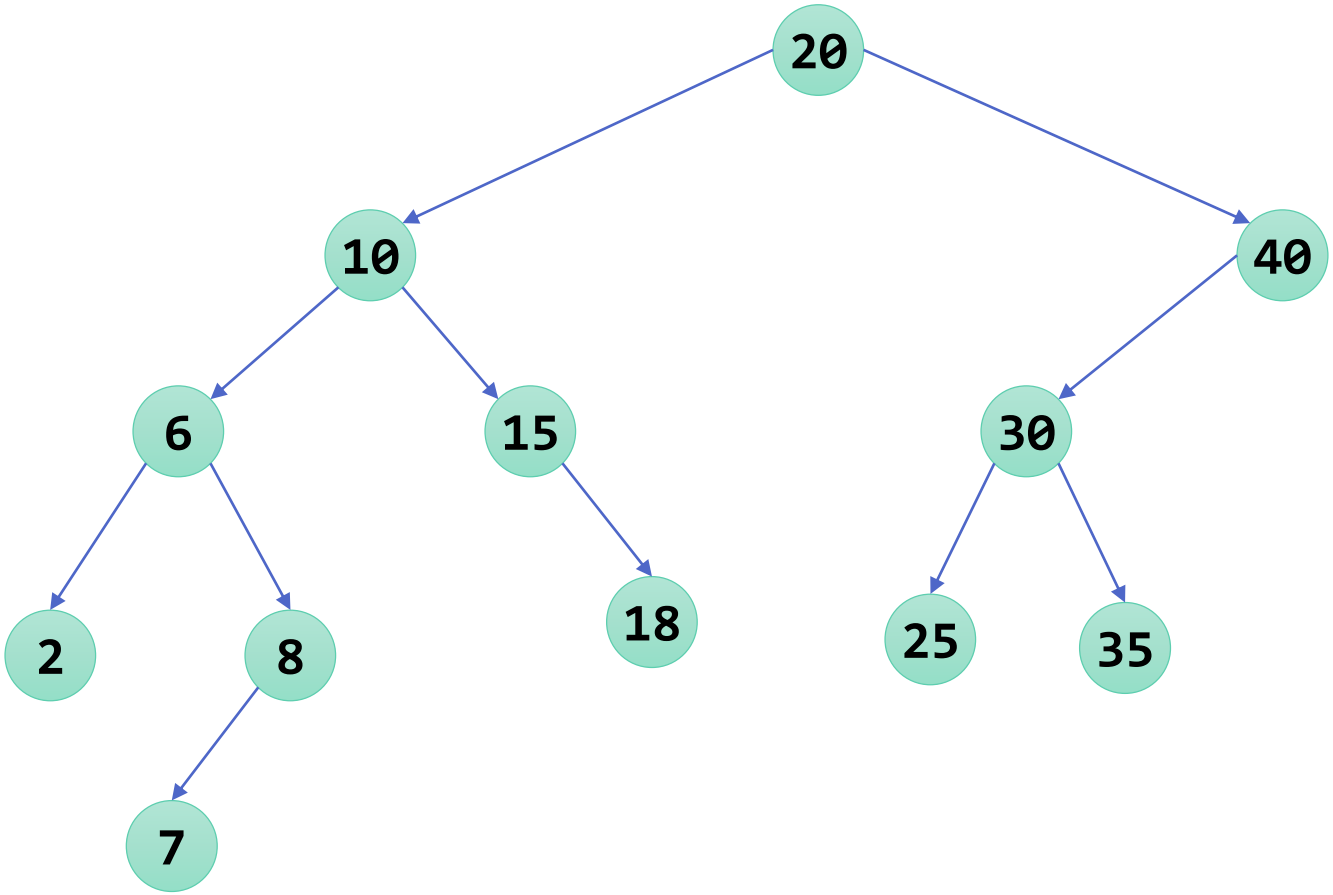
Insert a pair whose key is 7.

The Operation insert()



Insert a pair whose key is 18.

The Operation insert()

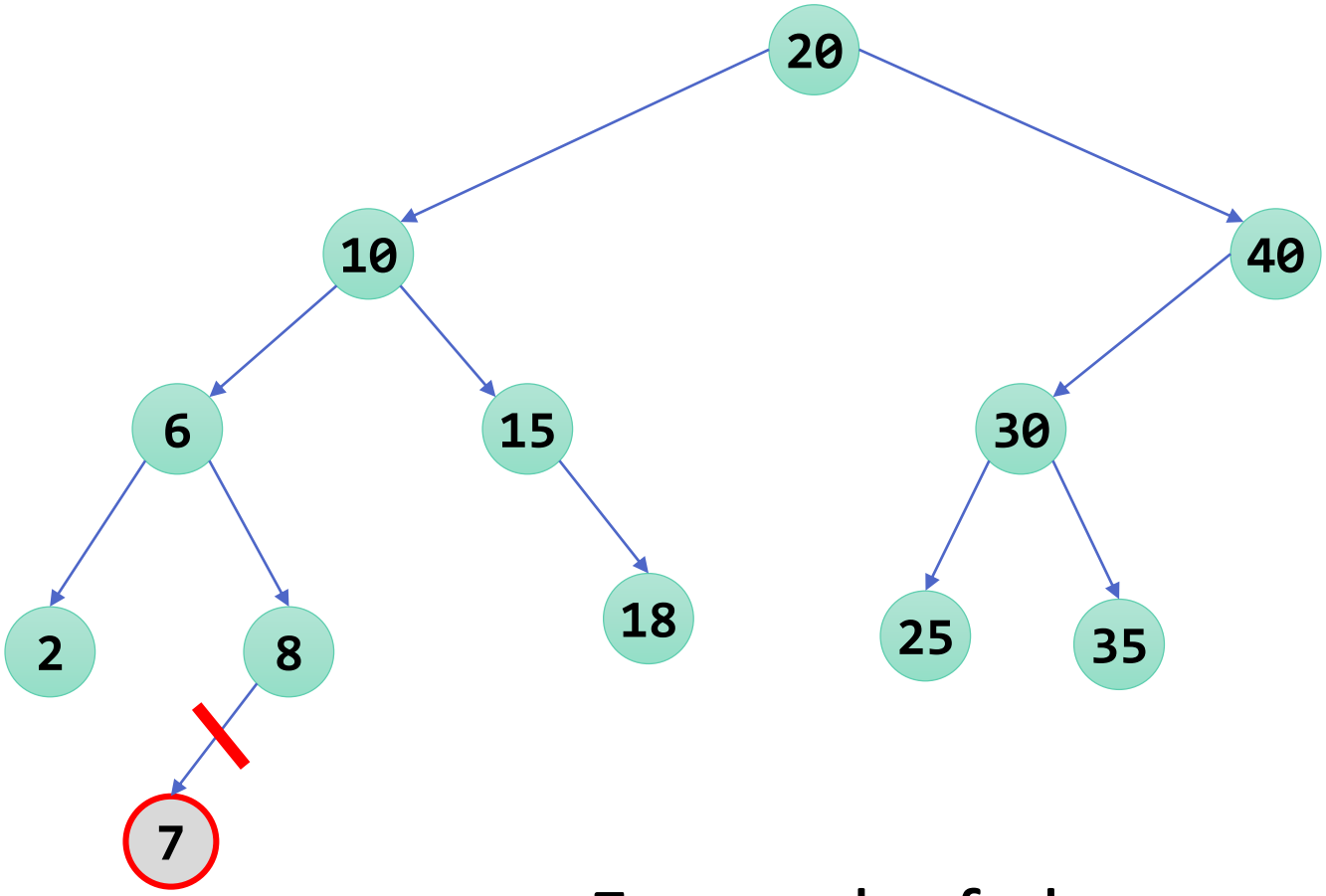


Complexity of insert() is O(height).

The Operation delete()

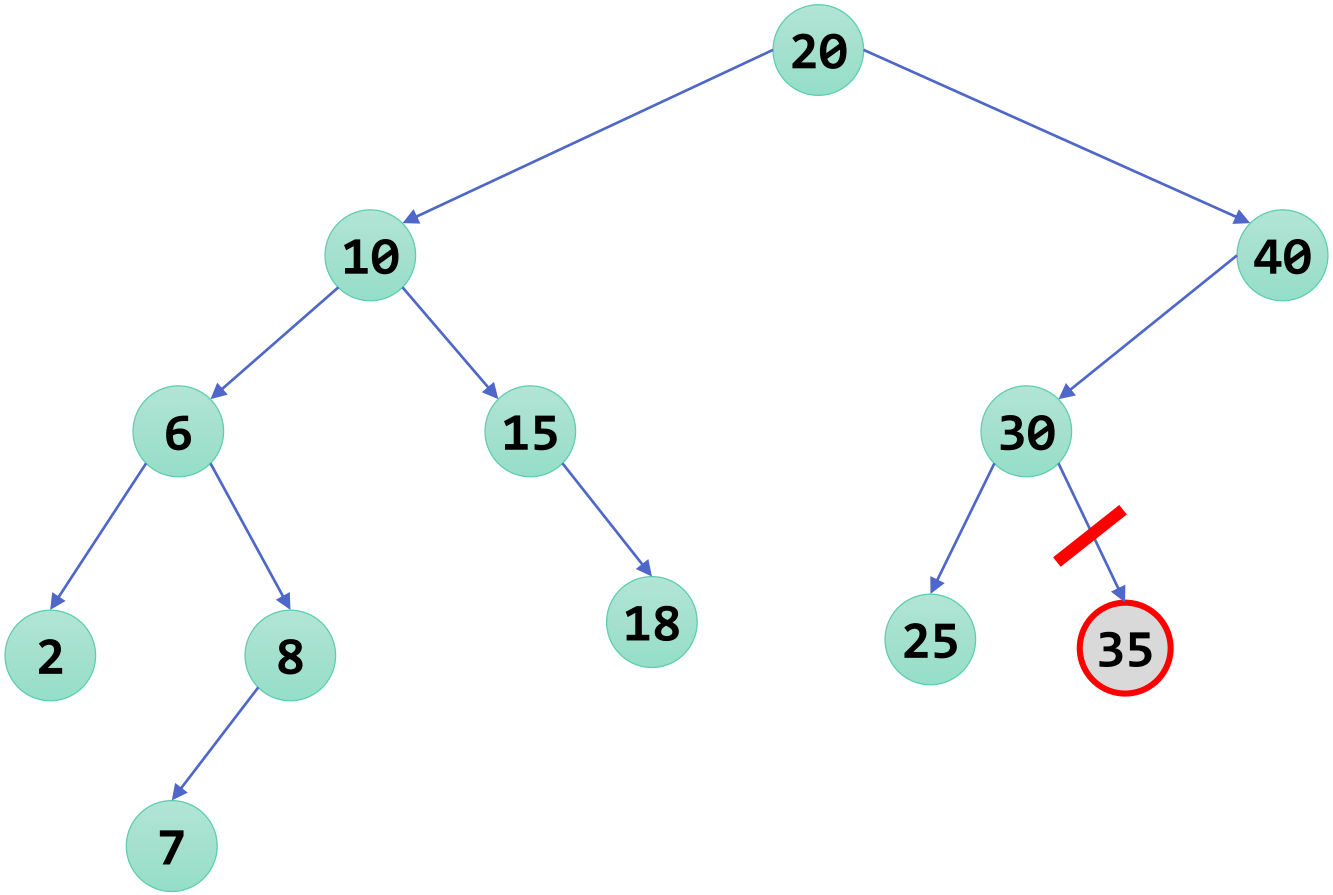
- Three cases:
  - Element is in a leaf.
  - Element is in a degree 1 node.
  - Element is in a degree 2 node.

Delete From A Leaf



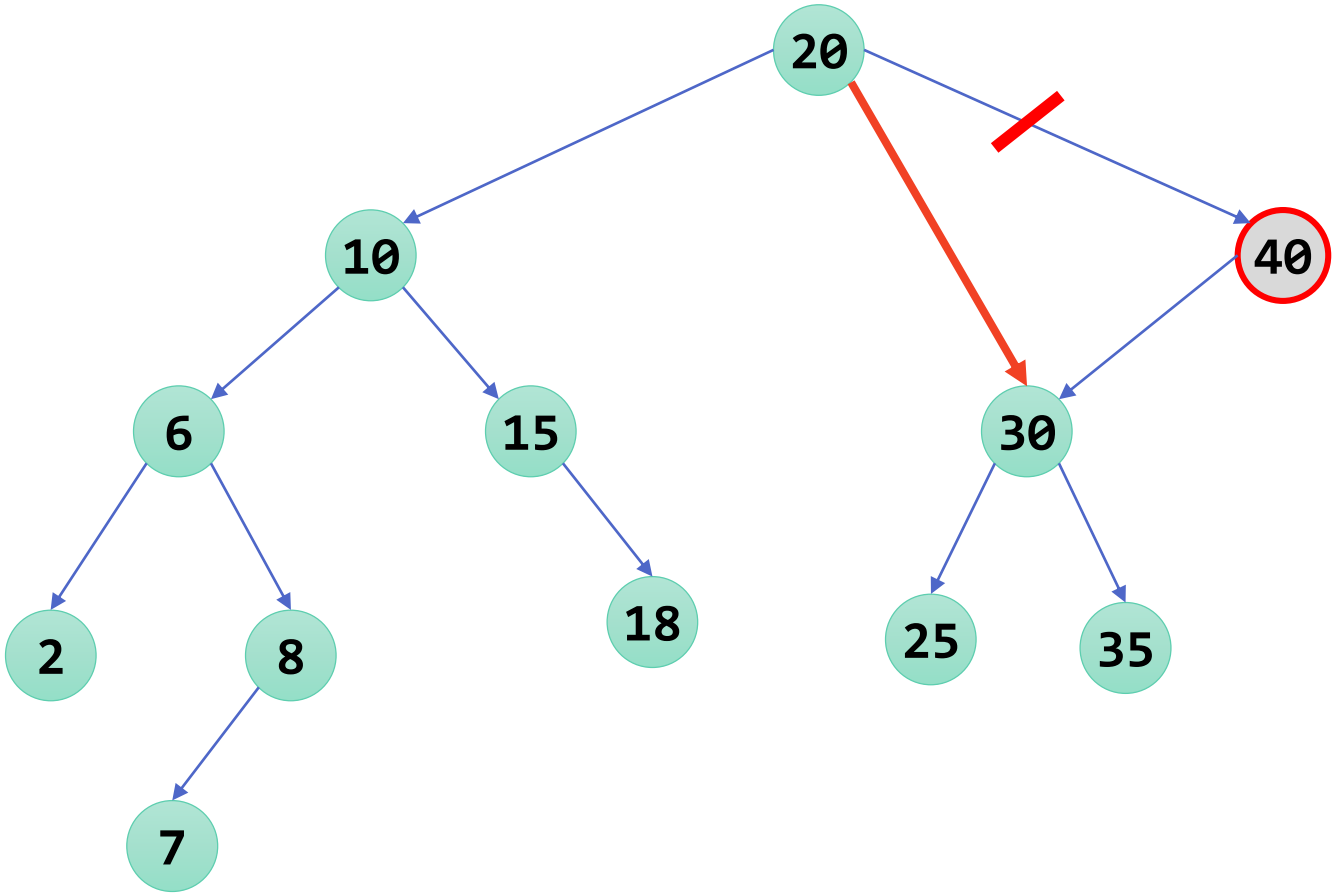
Erase a leaf element. key = 7

Delete From A Leaf



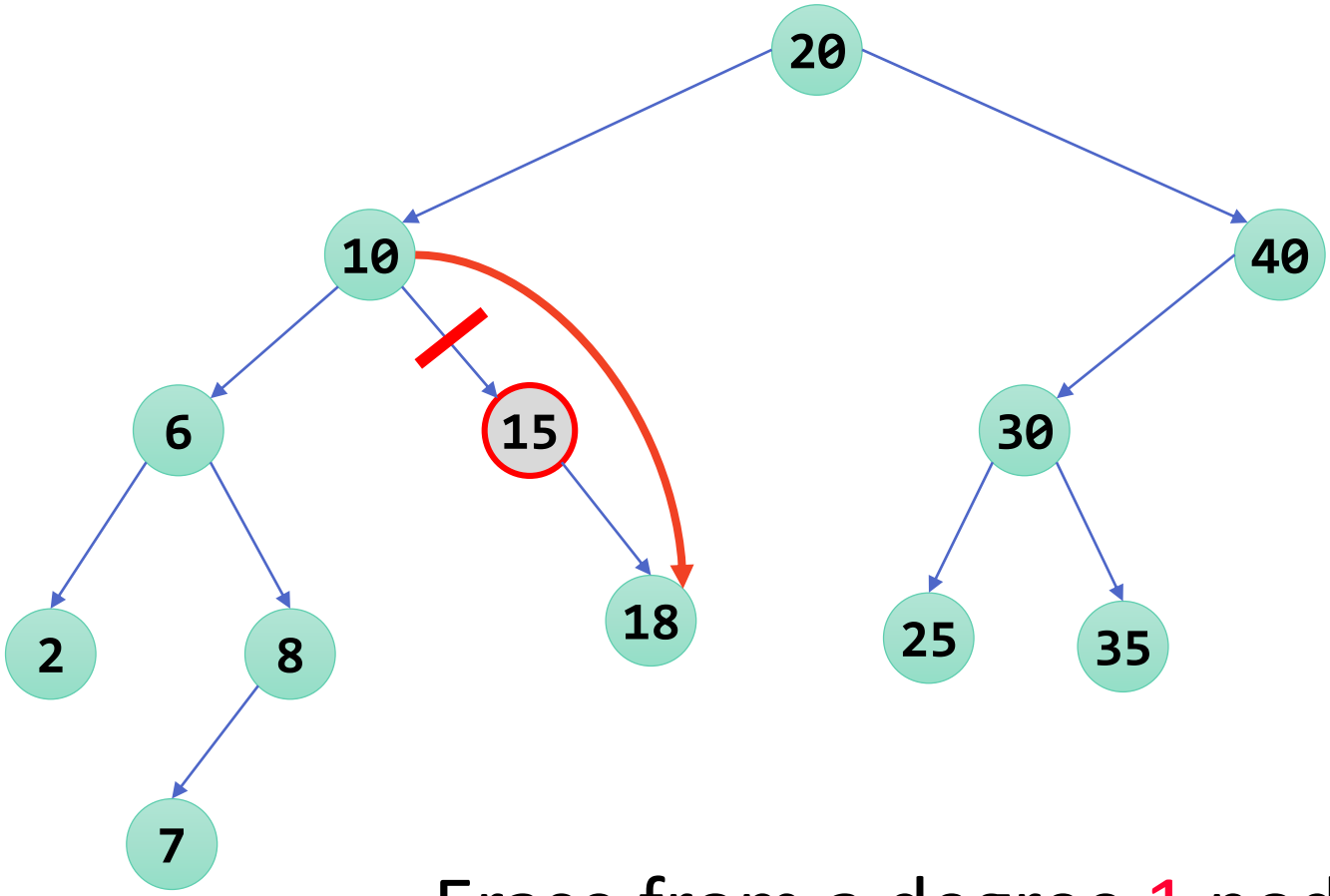
Erase a leaf element. key = 35

Delete From Degree 1 Node



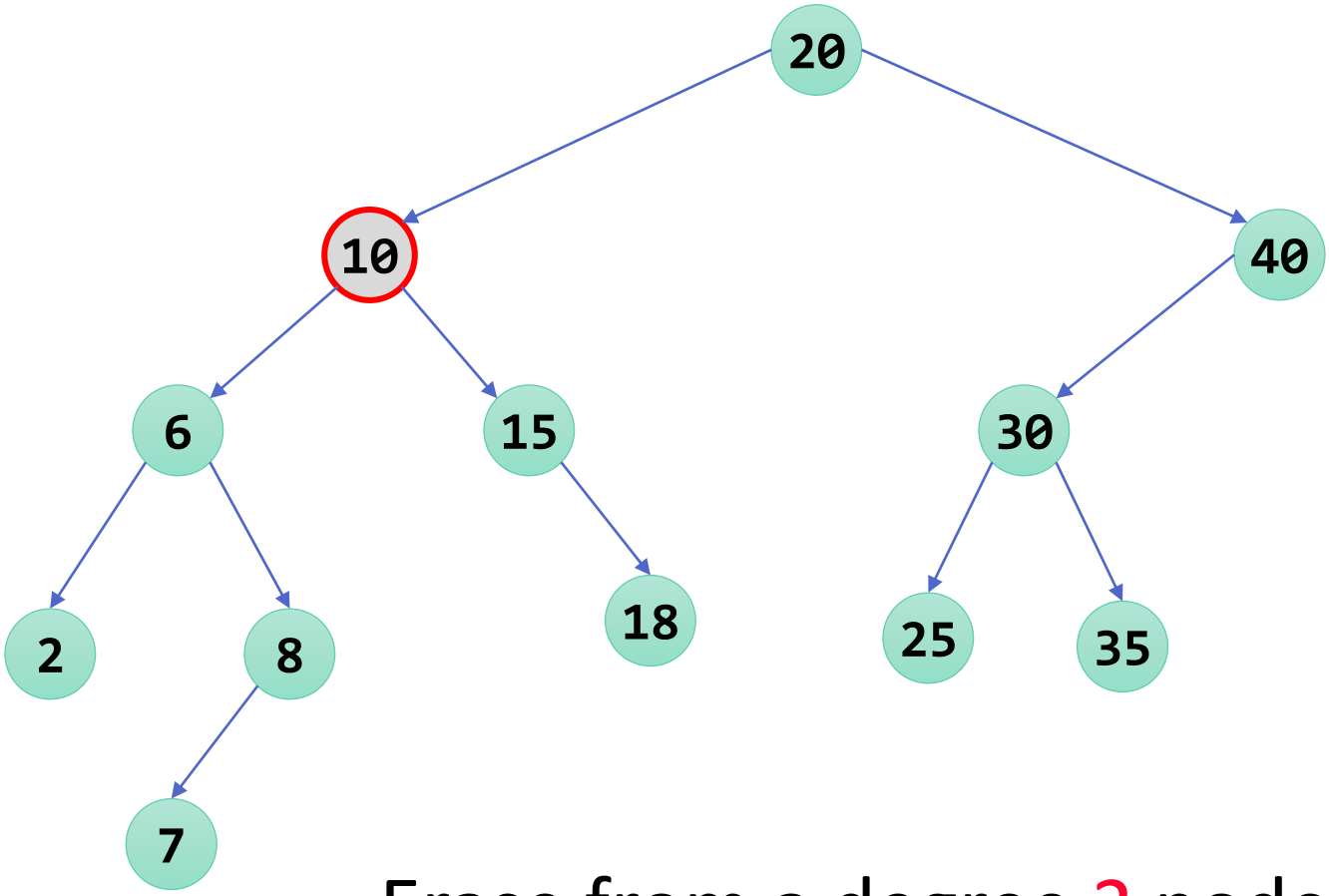
Erase from a degree 1 node. key = 40

Delete From Degree 1 Node



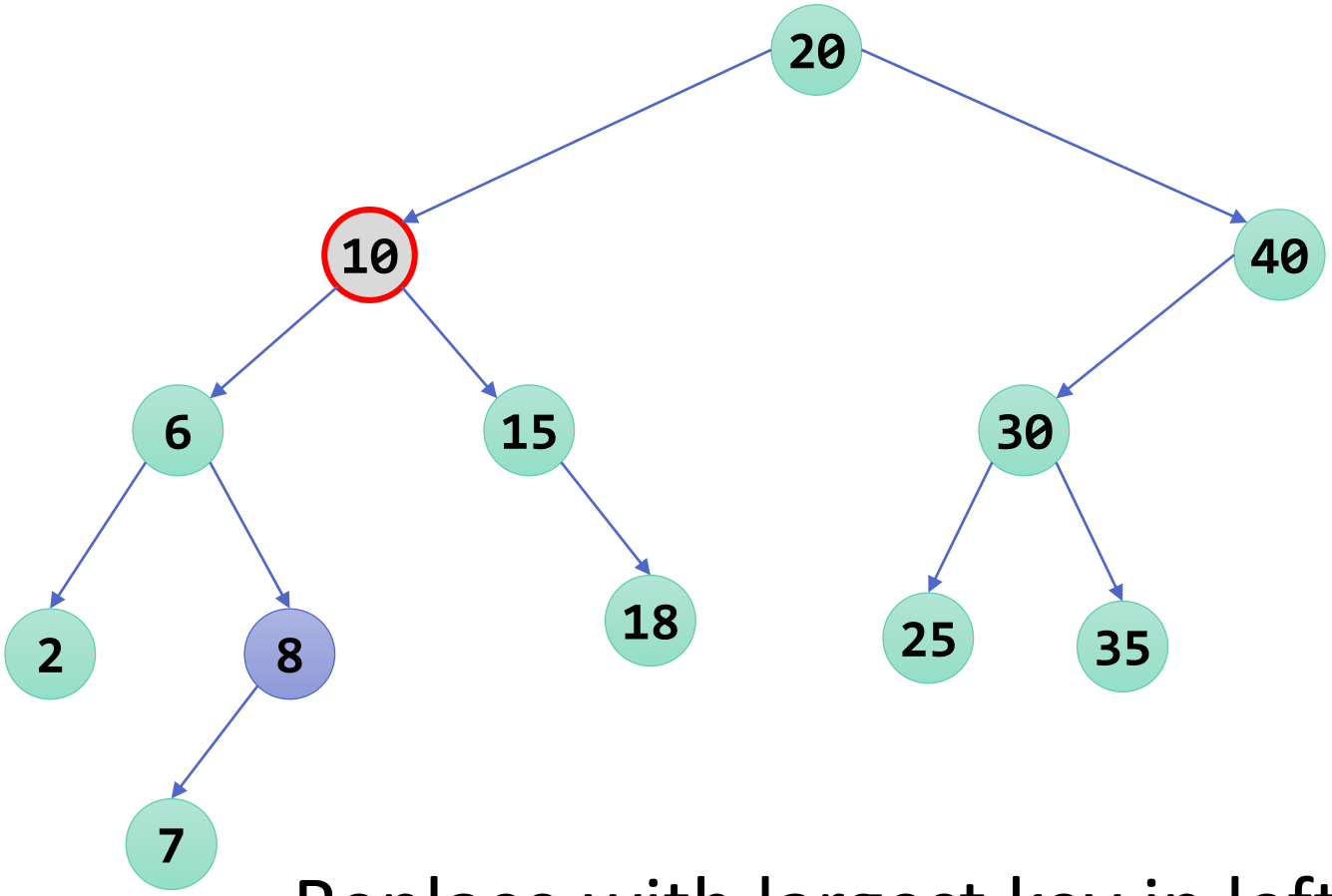
Erase from a degree 1 node. key = 15

Delete From Degree 2 Node



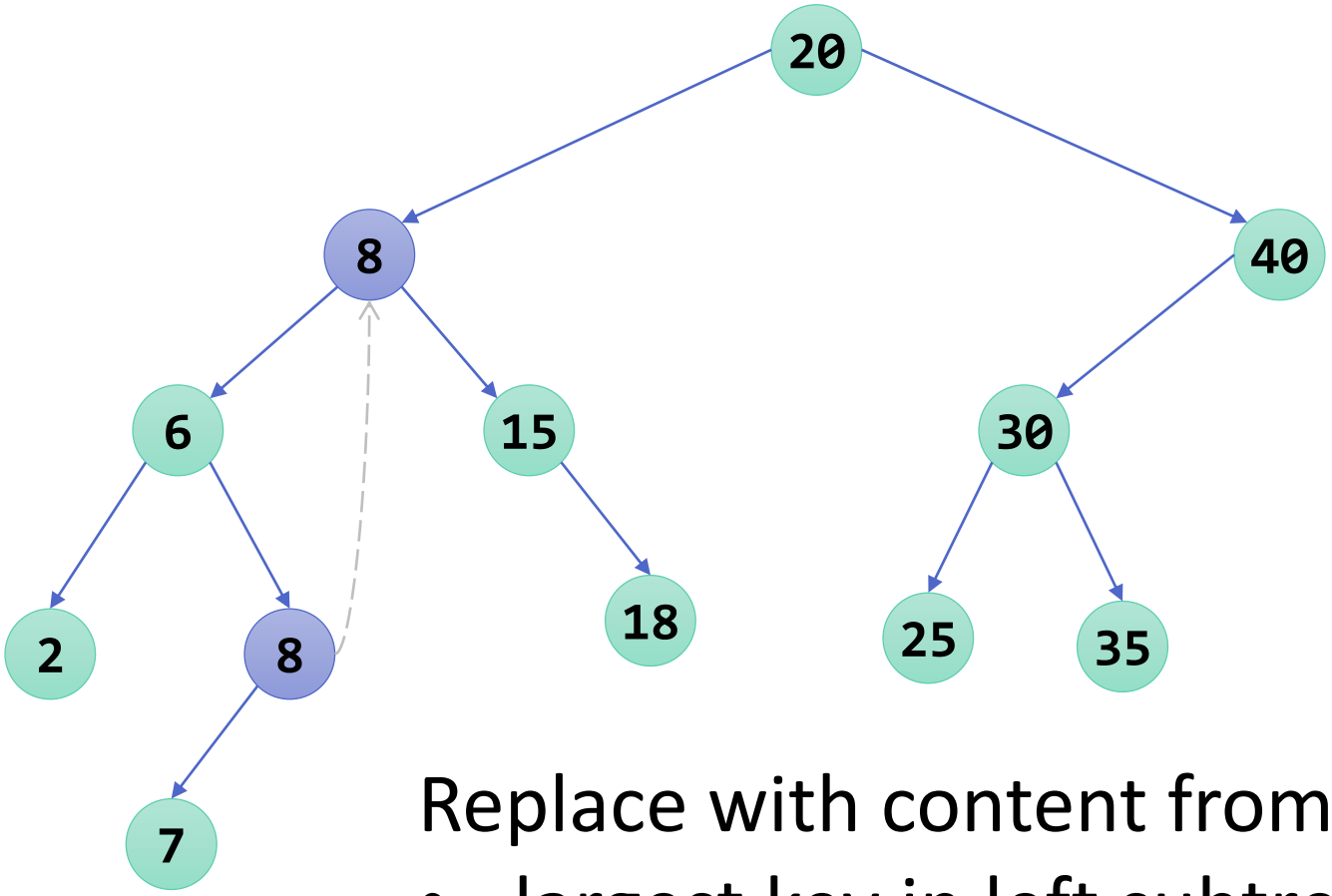
Erase from a degree 2 node. key = 10

Delete From Degree 2 Node



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

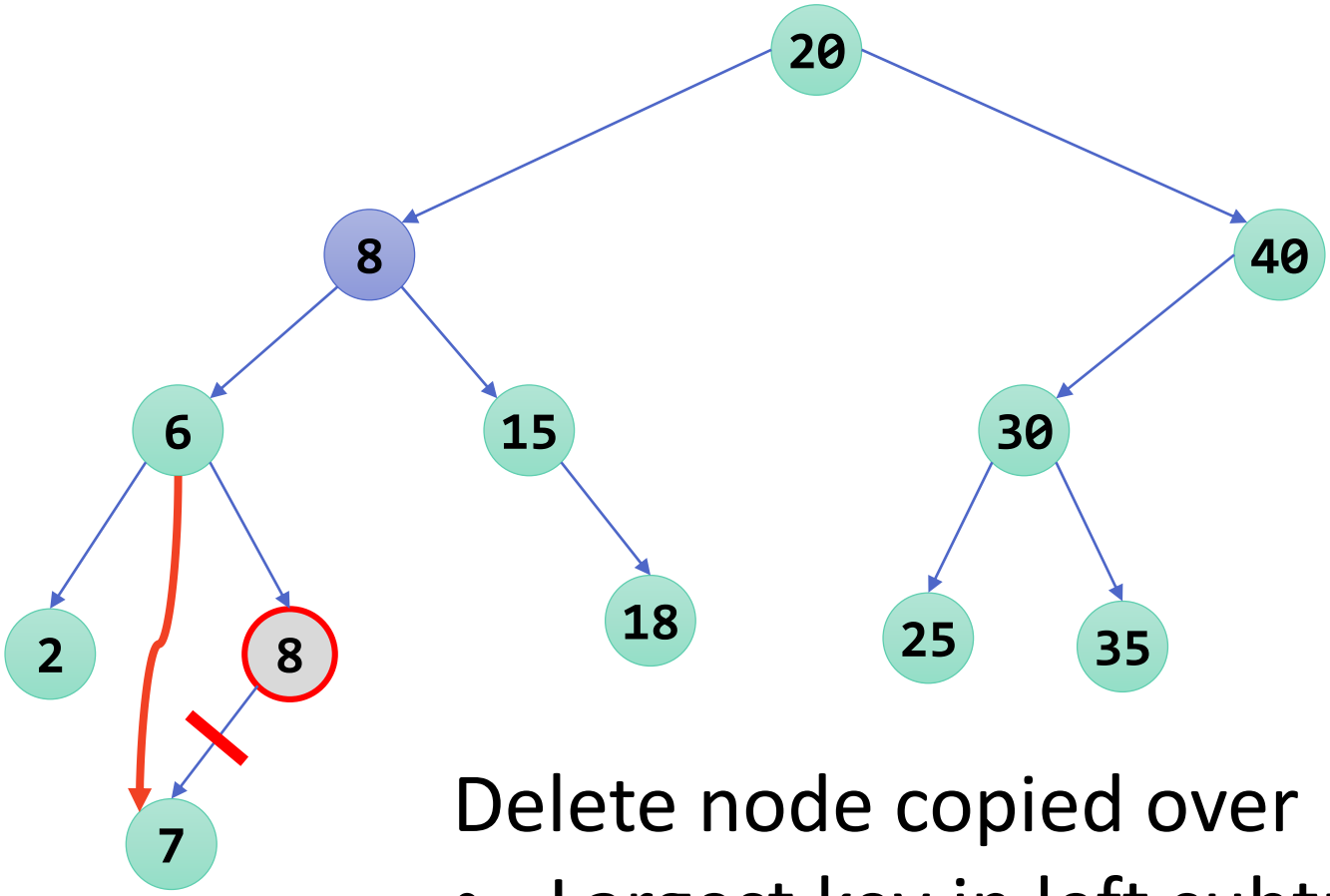
Delete From Degree 2 Node



Replace with content from

- largest key in left subtree, or
- smallest in right subtree

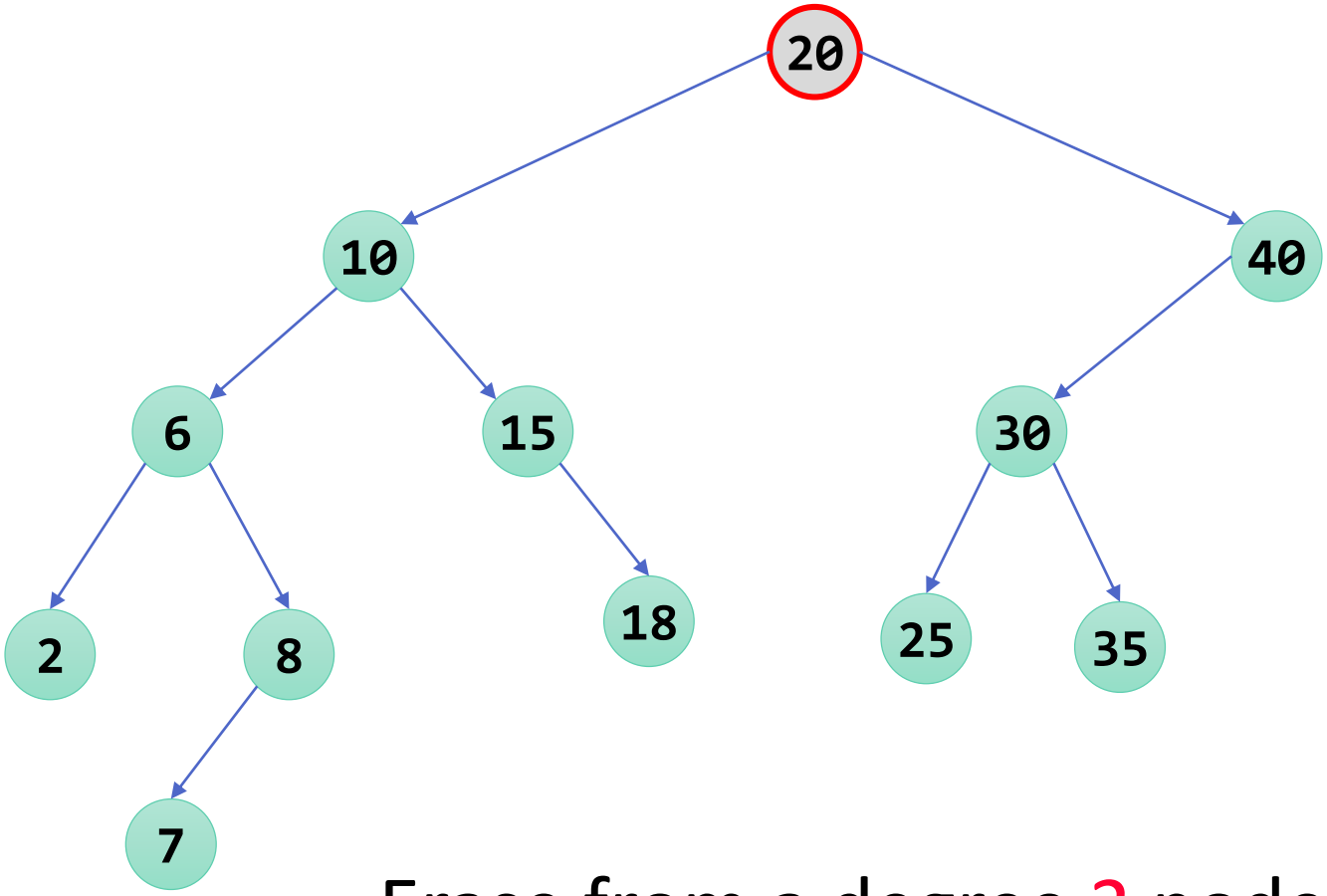
Delete From Degree 2 Node



Delete node copied over

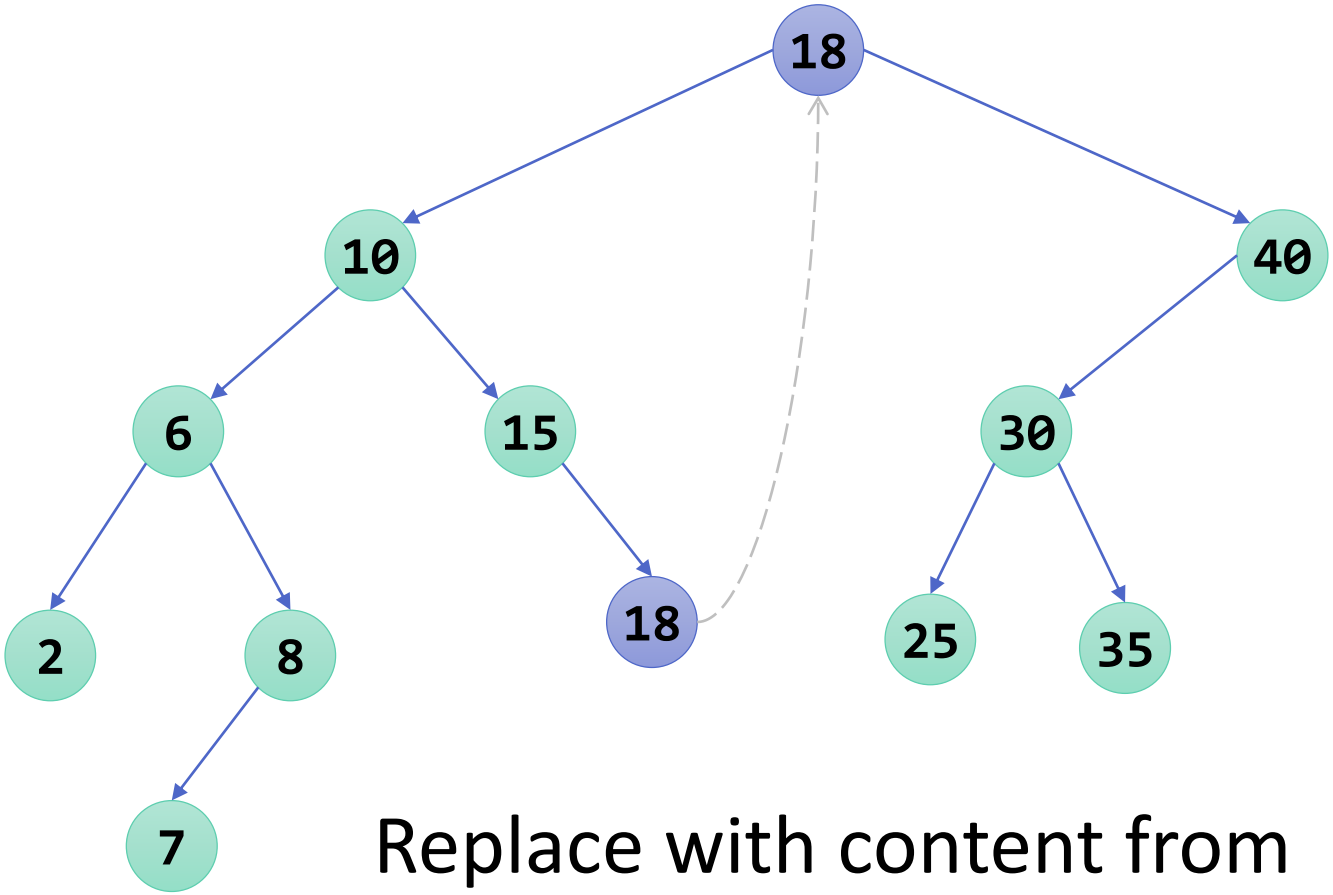
- Largest key in left subtree will be a leaf, or degree 1 node.

Delete From Degree 2 Node



Erase from a degree 2 node. key = 20

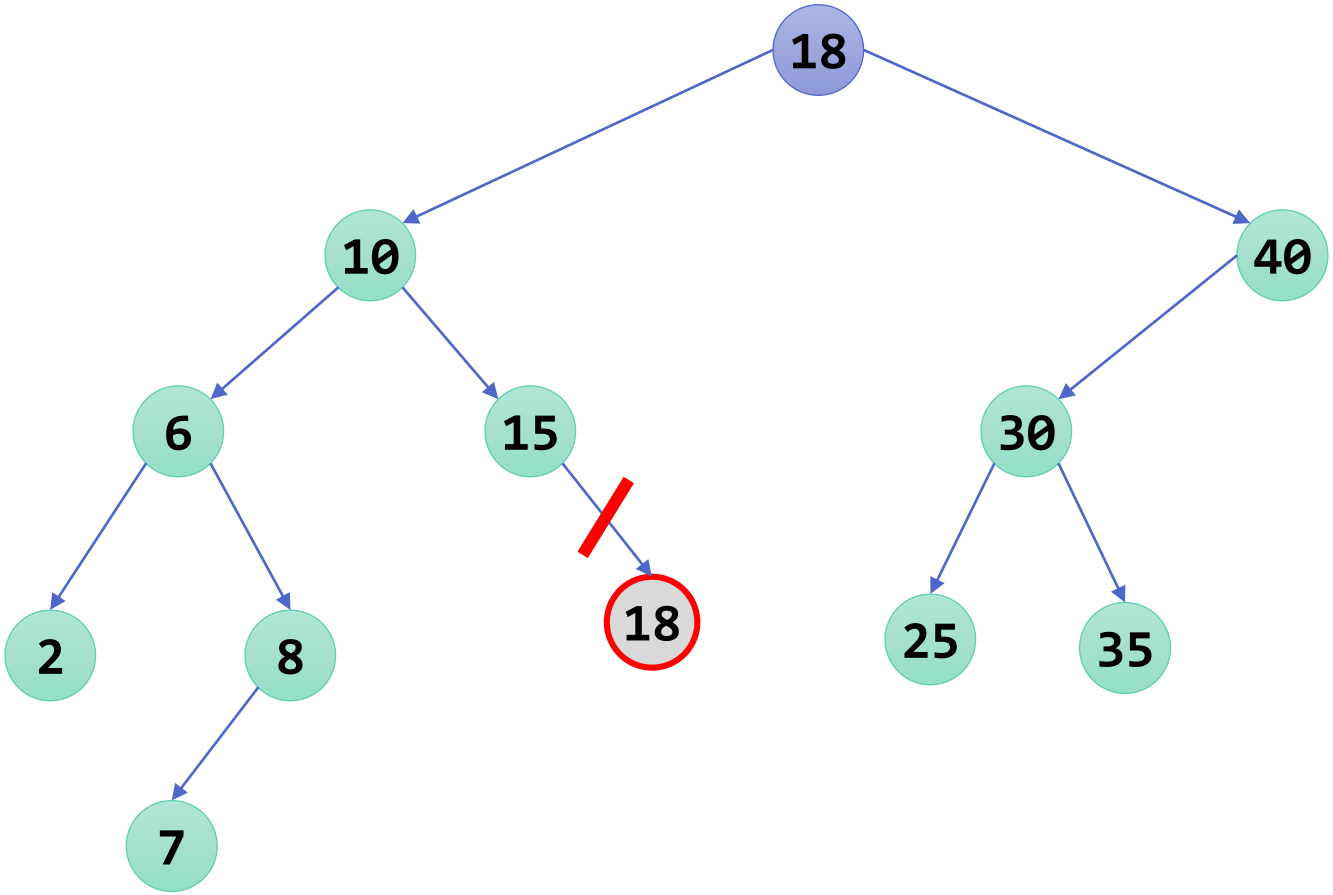
Delete From Degree 2 Node



Replace with content from

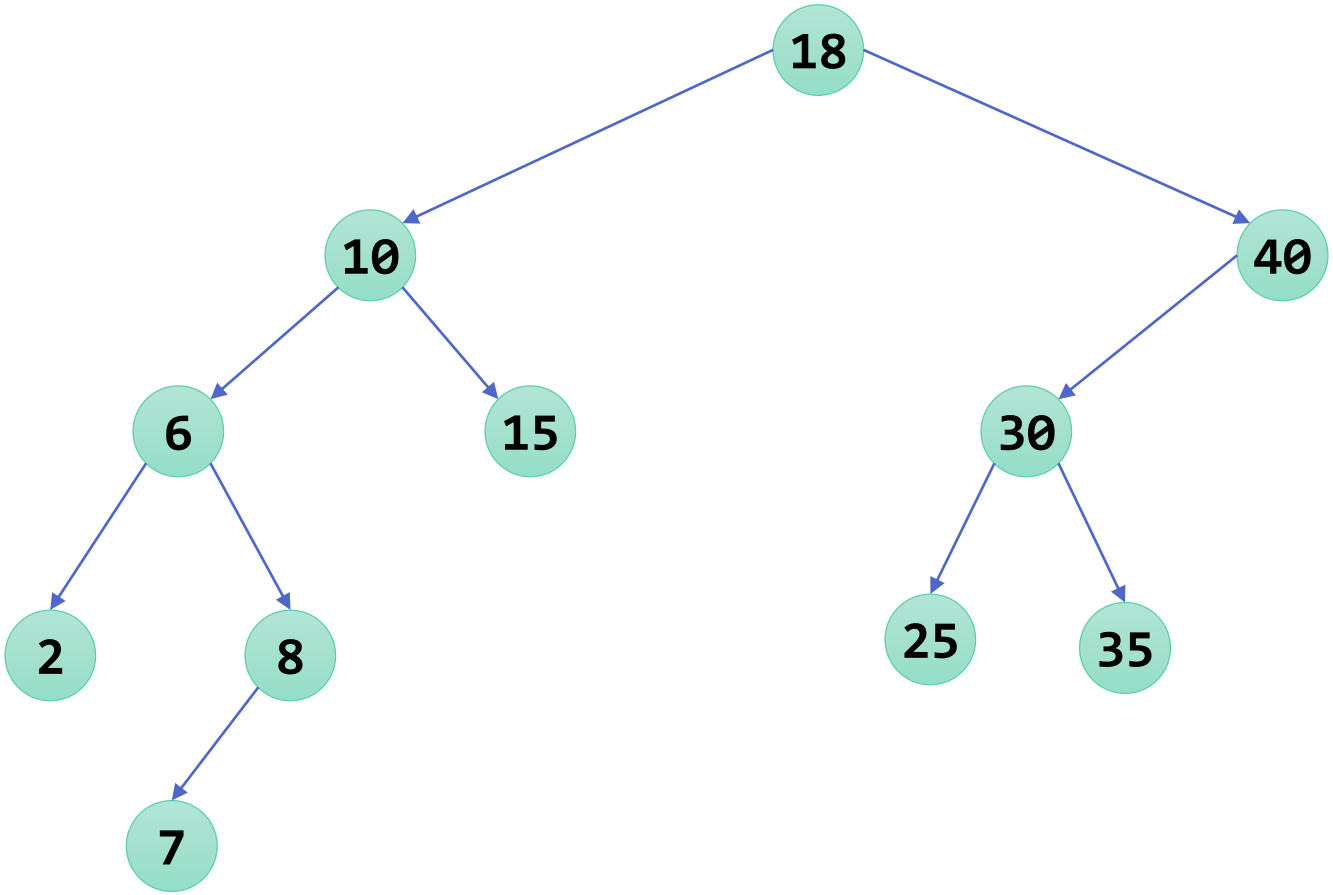
- largest key in left subtree, or
- smallest in right subtree

Delete From Degree 2 Node



Delete node copied over

Delete From Degree 2 Node



Complexity is O(height)

Tree Imbalances

- Inserting and Deleting in specific orders can cause tree to be imbalanced
  - E.g. insert in sorted acsending/descending order
  - Height of left and right subtrees are very different, skewed
- Causes complexity to tend to O(n) rather than O(log(n))
- Periodically *rebalance* if skew greater than a threshold
  - More after midterm exams

Tasks

- Self study (Sahni Textbook)
  - **Read**: Chapter 11.0-11.6, Trees & Binary Trees from textbook
  - **Try:** Pseudo-code for BST operations (find, insert, delete)
- ~~Finish Assignment 3 by Wed Sep 28 *(75 points)*~~
  - Late submissions from Thu-Sun will entail **5 points penalty per day**
  - Submissions on or after Mon, Oct 3 will not be accepted
- 30 Sep (Fri) Institute holiday. We will have class at 10am.
- Move Midterm from Oct 5 to Oct 7
  - All lectures till Trees & Searching will be in syllabus

Questions?