Data Structures
Binary Search Trees 2

CS 225
Brad Solomon & G Carl Evans

September 20, 2023

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Department of Computer Science

Learning Objectives

Review binary search trees

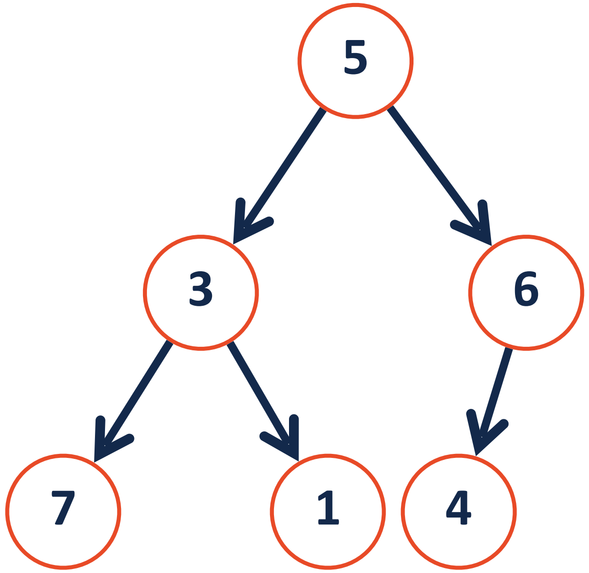
Continue implementing BST ADT

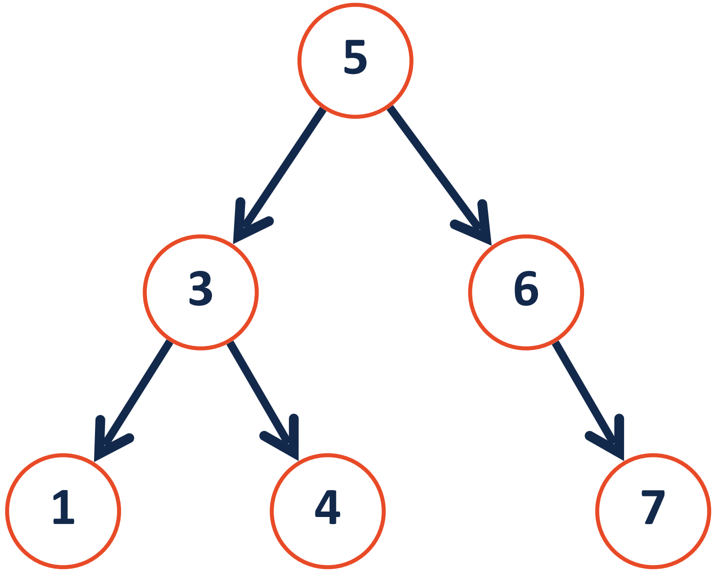
Discuss pros and cons of BST (and possible improvements)

Improved search on a binary tree

5	3	6	7	1	4

1	3	4	5	6	7





Dictionary ADT

Data is often organized into key/value pairs:

Word → Definition

Course Number → Lecture/Lab Schedule

Node → Incident Edges

Flight Number → Arrival Information

URL → HTML Page

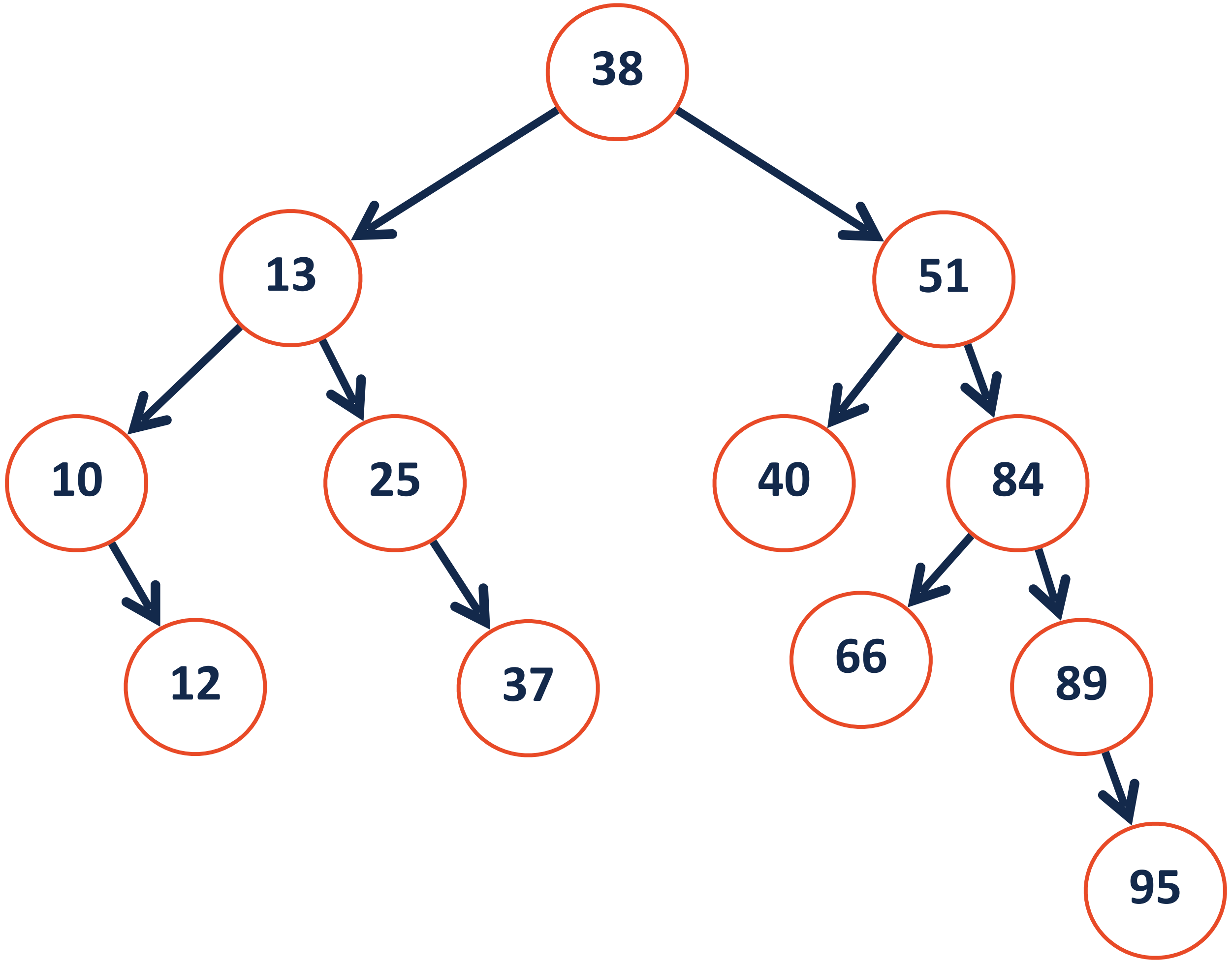
…

Binary Search Tree (BST)

A **BST** is a binary tree $T = TreeNode(val, T_L, T_r)$ such that:

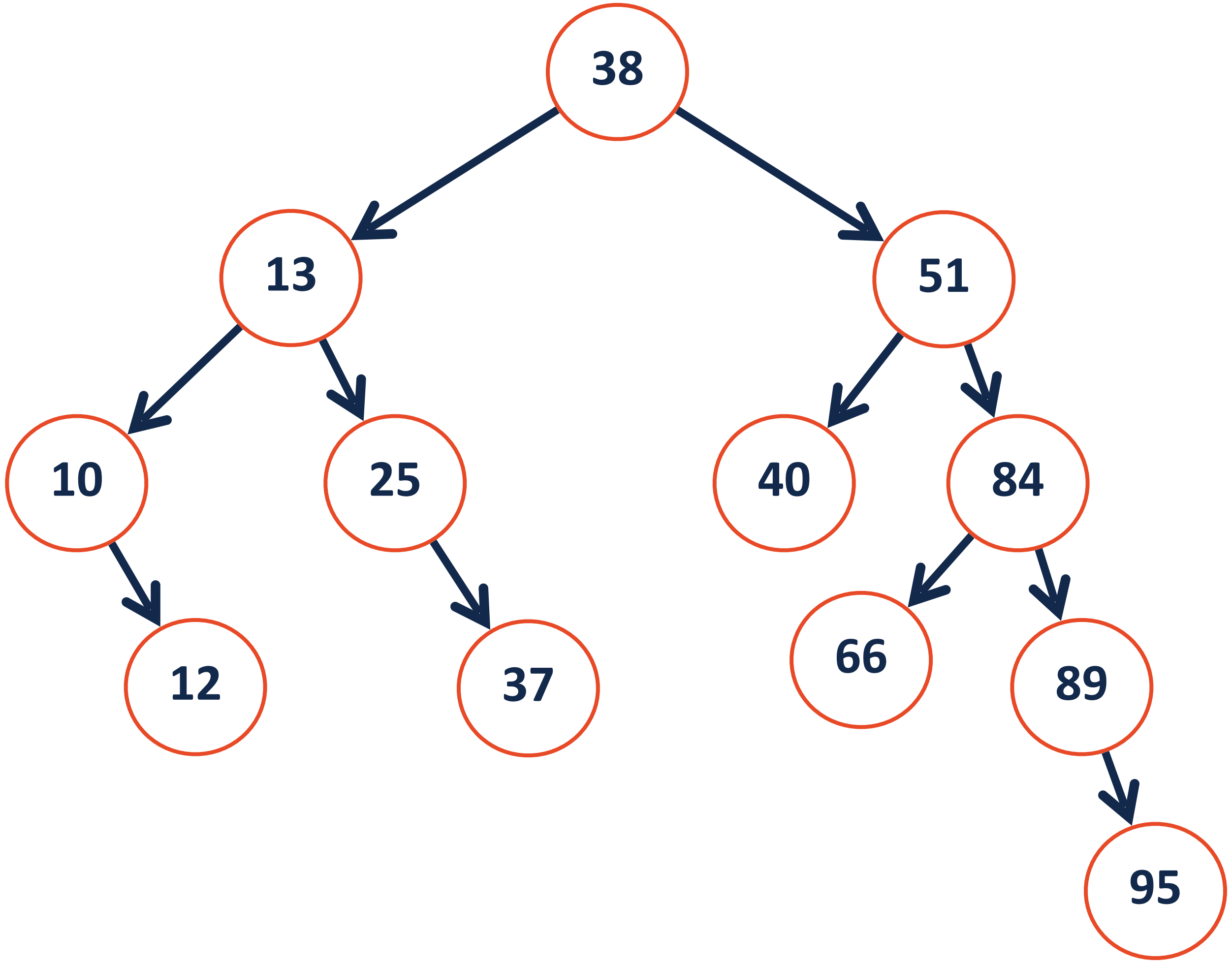
$\forall n \in T_L, n.val < T.val$

$\forall n \in T_R, n.val > T.val$



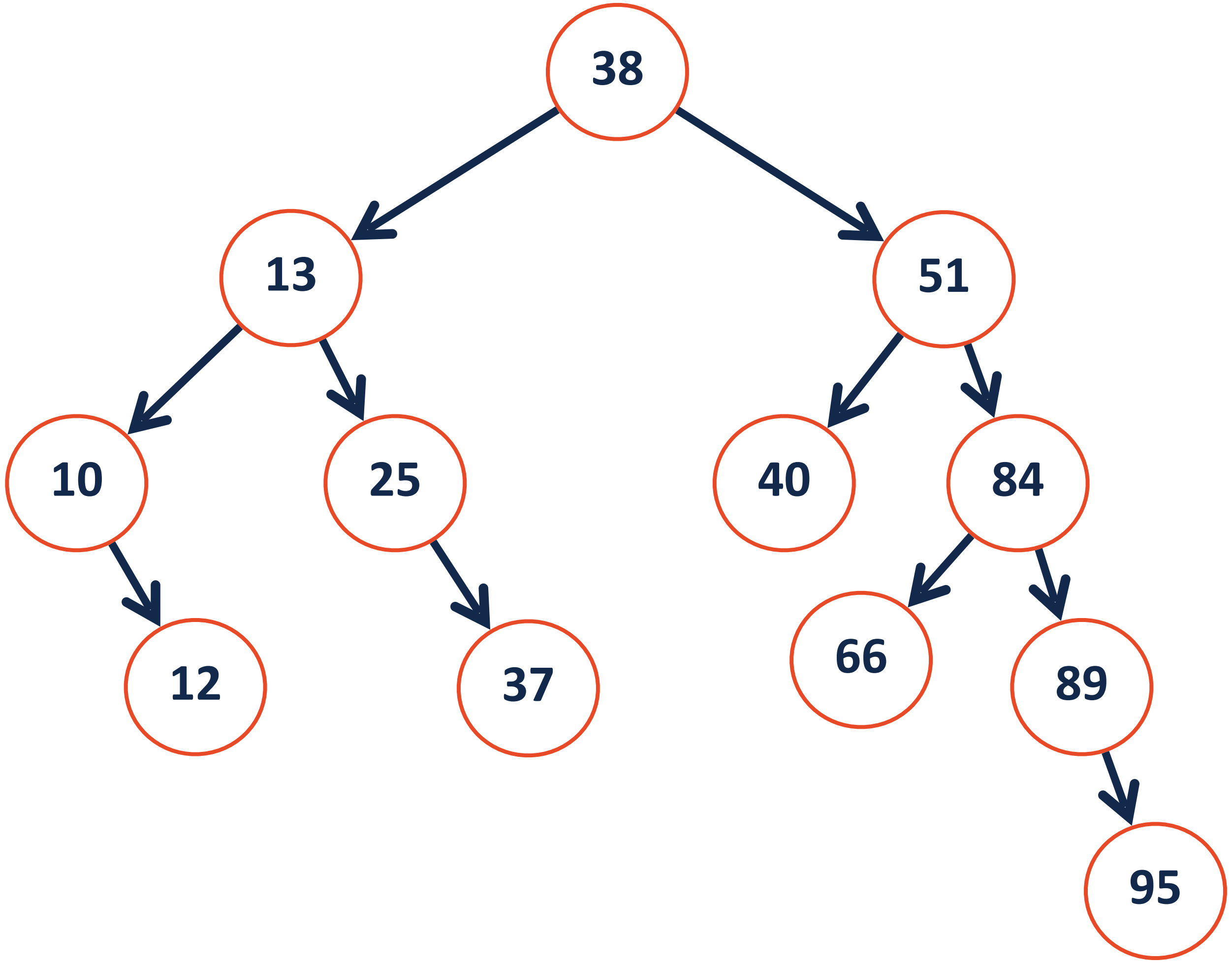
BST Remove

`remove(40)`



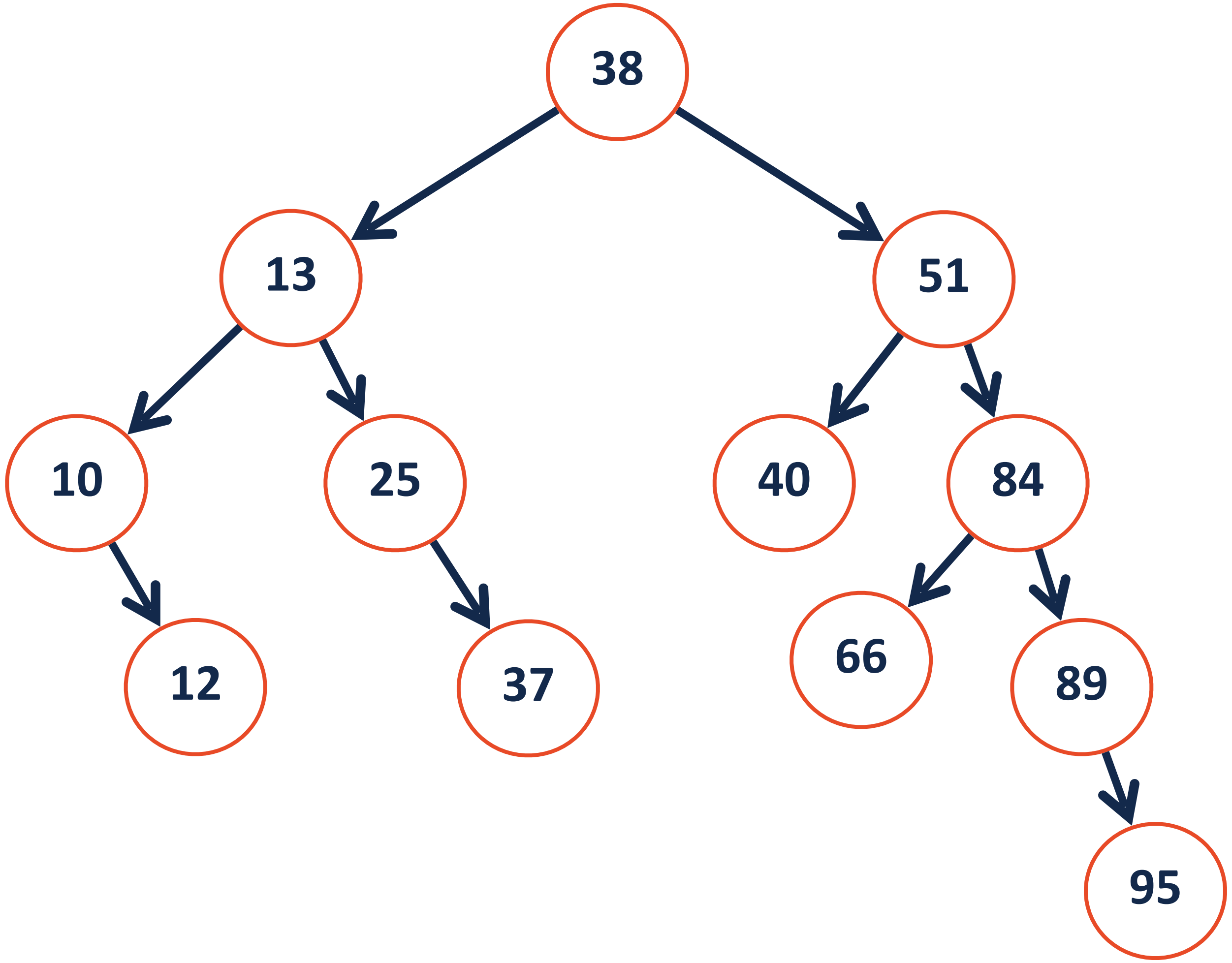
BST Remove

`remove(25)`



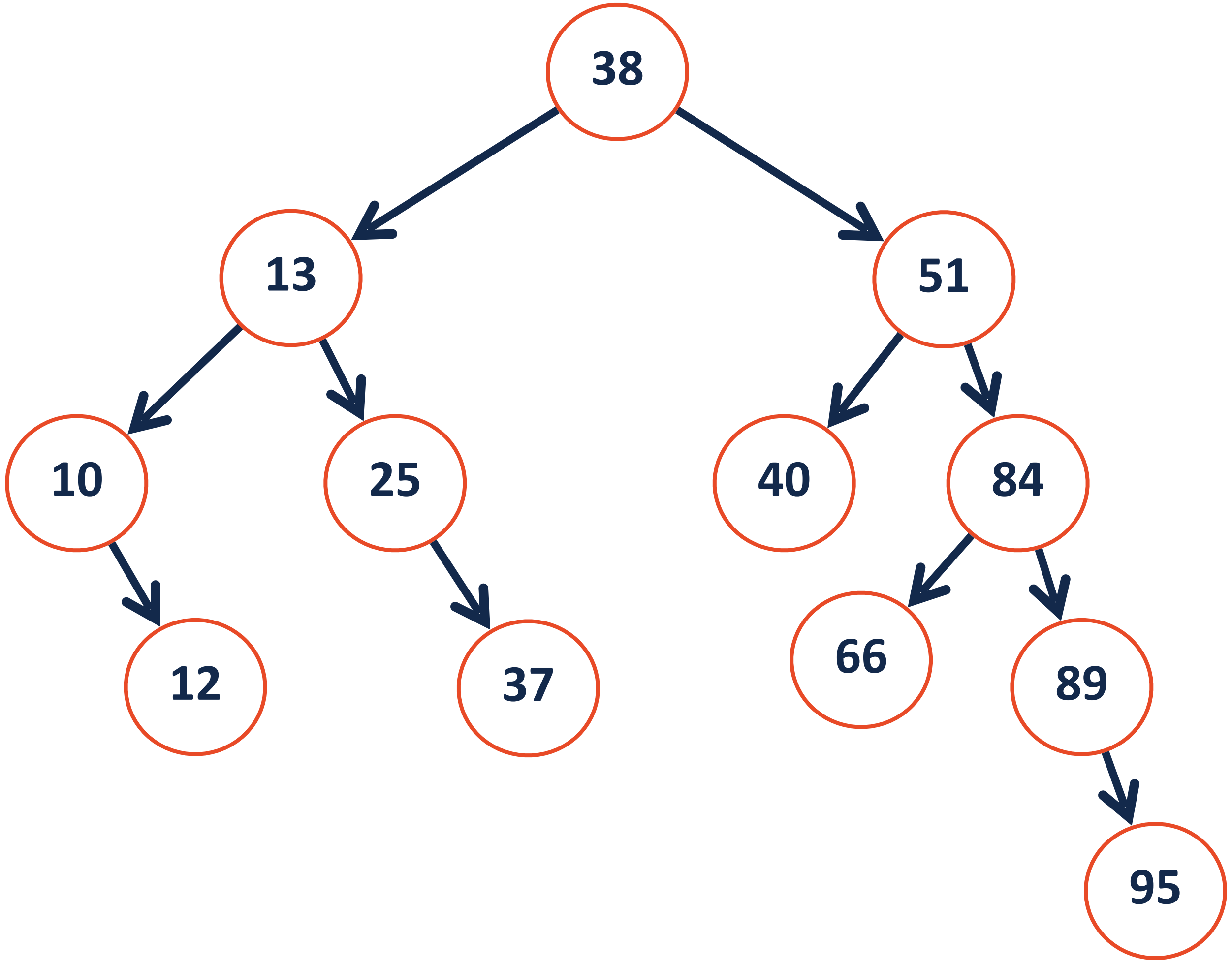
BST Remove

`remove(13)`

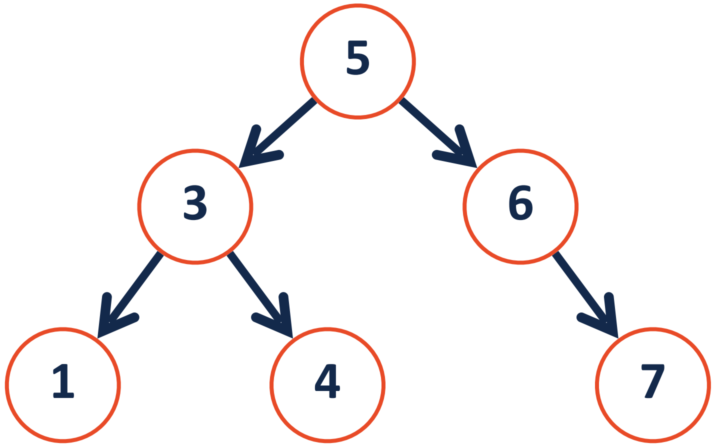


BST Remove

`remove(51)`

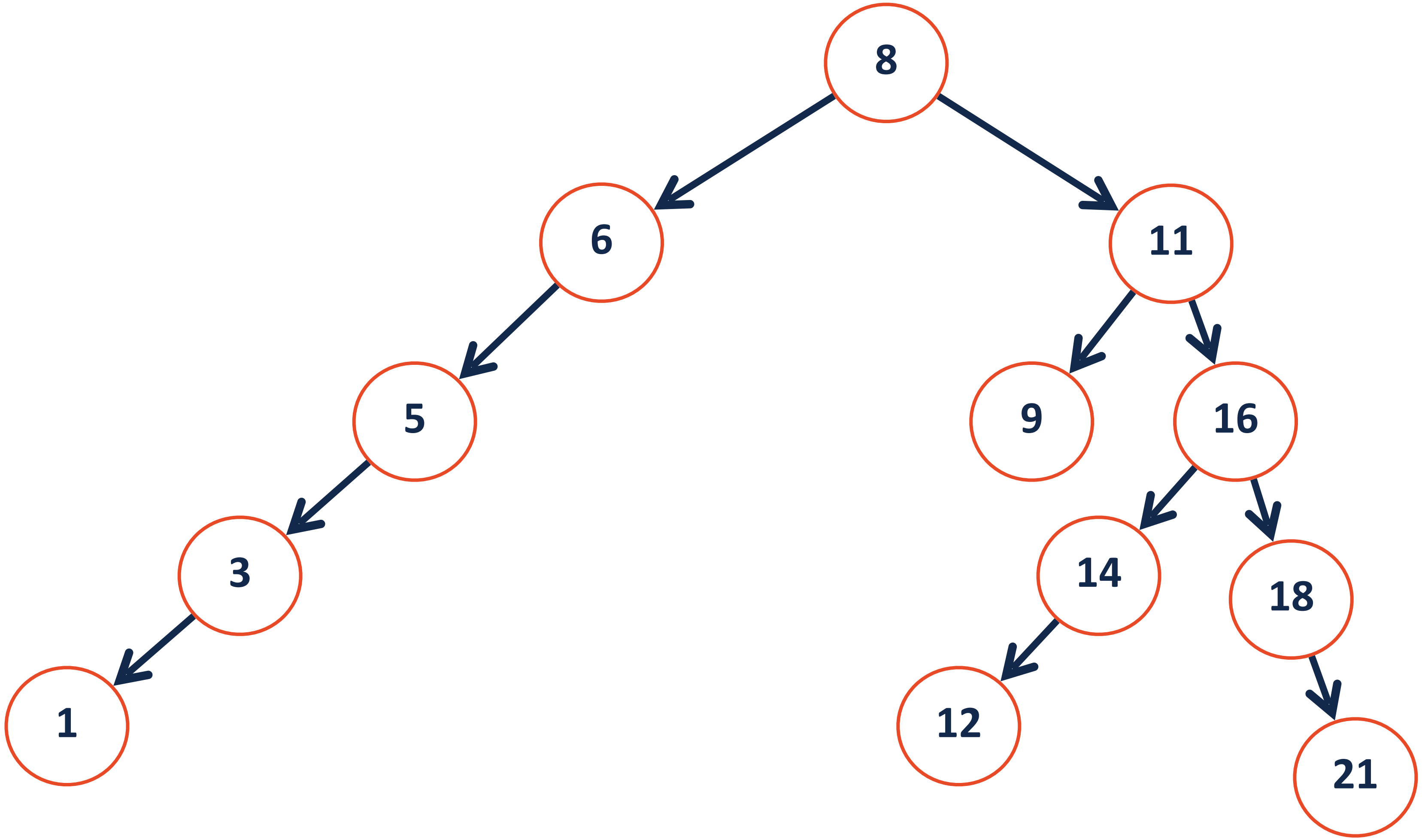


```
template<typename K, typename V>

void _remove(TreeNode *& root, const K & key) {



















}
```



BST Remove

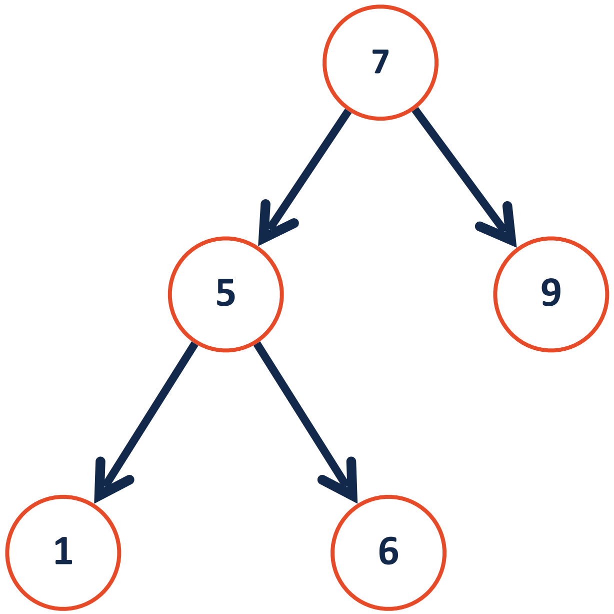
What will the tree structure look like if we remove node 16 using IOS?

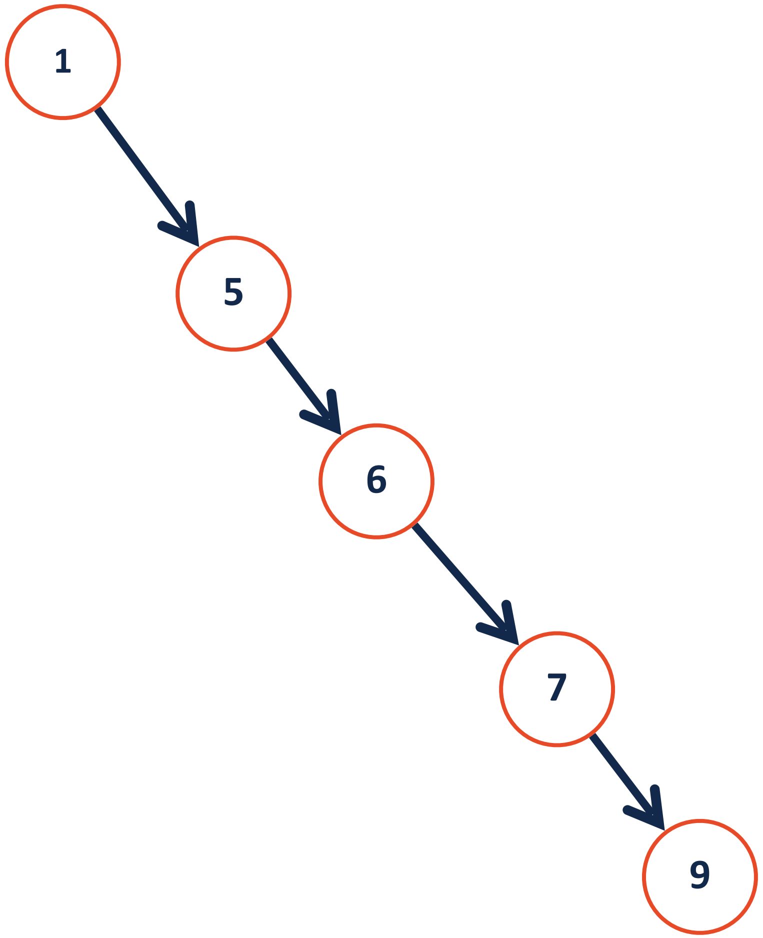


BST Analysis – Running Time

Operation	BST Worst Case
find	
insert	
remove	
traverse	

Limiting the height of a tree





Option A: Correcting bad insert order

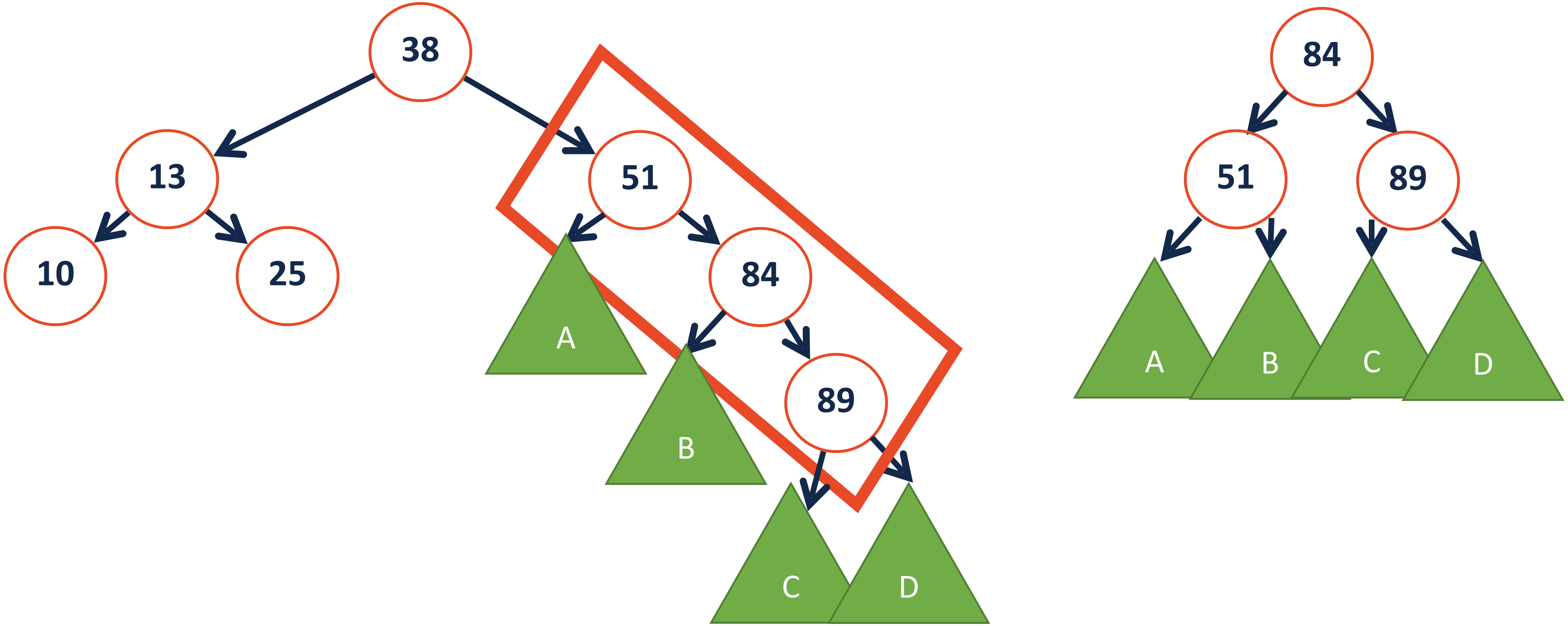
The height of a BST depends on the order in which the data was inserted

Insert Order: [1, 3, 2, 4, 5, 6, 7]

Insert Order: [4, 2, 3, 6, 7, 1, 5]

AVL-Tree: A self-balancing binary search tree

Rather than fixing an insertion order, just correct the tree as needed!


38
13
51
10
25
84
A
89
B
C
D
84
51
89
A
B
C
D