Binary Search Trees: Balance

Daniel Kane

Department of Computer Science and Engineering
University of California, San Diego

Data Structures
Data Structures and Algorithms

Learning Objectives

- Think about the runtime of basic binary tree operations.
- Understand the motivation behind binary search tree balance.
- Implement a rotation.

Outline

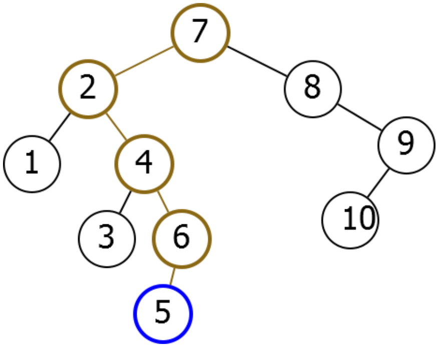
1. Runtime
2. Balanced Trees
3. Rotations

Runtime

How long do Binary Search Tree operations take?

Find

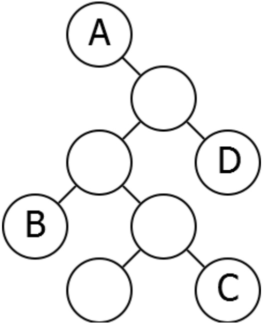
`Find(5)`



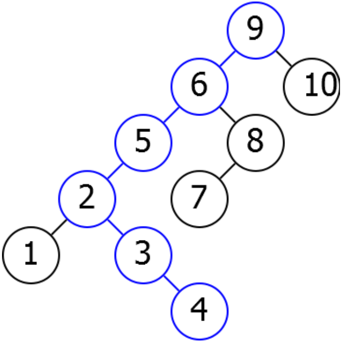
Number of operations = $O(\text{Depth})$

Problem

Which nodes will be faster to search for in the following tree?



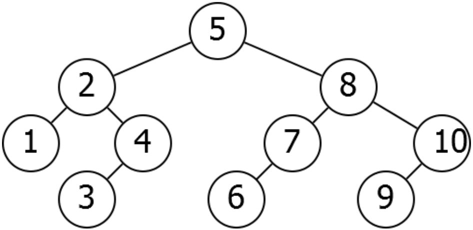
Example I



Depth can be as bad as n.

Outline

Example II



Depth can be much smaller.

Balance

- Want left and right subtrees to have approximately the same size.

Balance

- Want left and right subtrees to have approximately the same size.
- Suppose perfectly balanced:

Balance

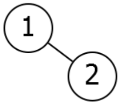
- Want left and right subtrees to have approximately the same size.
- Suppose perfectly balanced:
  - Each subtree half the size of its parent.
  - After $\log_2(n)$ levels, subtree of size 1.
  - Operations run in $O(\log(n))$ time.

Problem

Insertions and deletions can destroy balance!

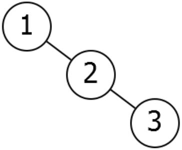
Problem

Insertions and deletions can destroy balance!



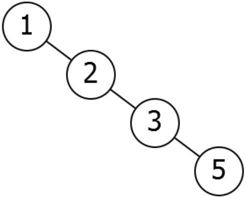
Problem

Insertions and deletions can destroy balance!



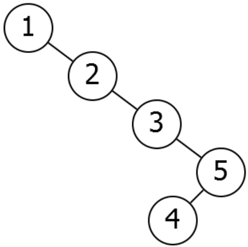
Problem

Insertions and deletions can destroy balance!



Problem

Insertions and deletions can destroy balance!



Outline

1. Runtime
2. Balanced Trees
3. Rotations

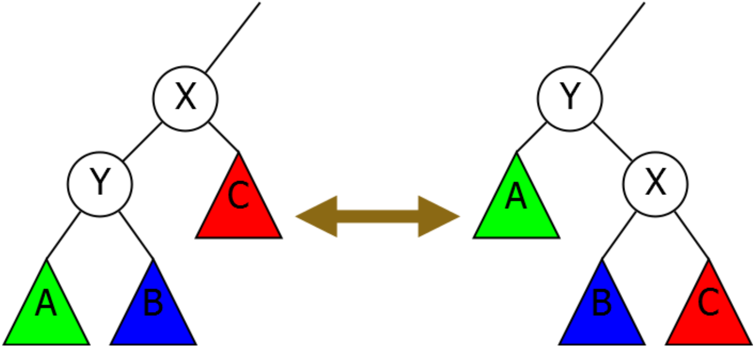
Rebalancing

Idea: Rearrange tree to maintain balance.

Rebalancing

Idea: Rearrange tree to maintain balance.
Problem: How do we rearrange tree while maintaining order?

Rotations



$$A < Y < B < X < C$$

Implementation

RotateRight(X)

$P \leftarrow X.\texttt{Parent}$
$Y \leftarrow X.\texttt{Left}$
$B \leftarrow Y.\texttt{Right}$
$Y.\texttt{Parent} \leftarrow P$
$P.\texttt{AppropriateChild} \leftarrow Y$
$X.\texttt{Parent} \leftarrow Y, Y.\texttt{Right} \leftarrow X$
$B.\texttt{Parent} \leftarrow X, X.\texttt{Left} \leftarrow B$

Next Time

How to keep a tree balanced. AVL trees.