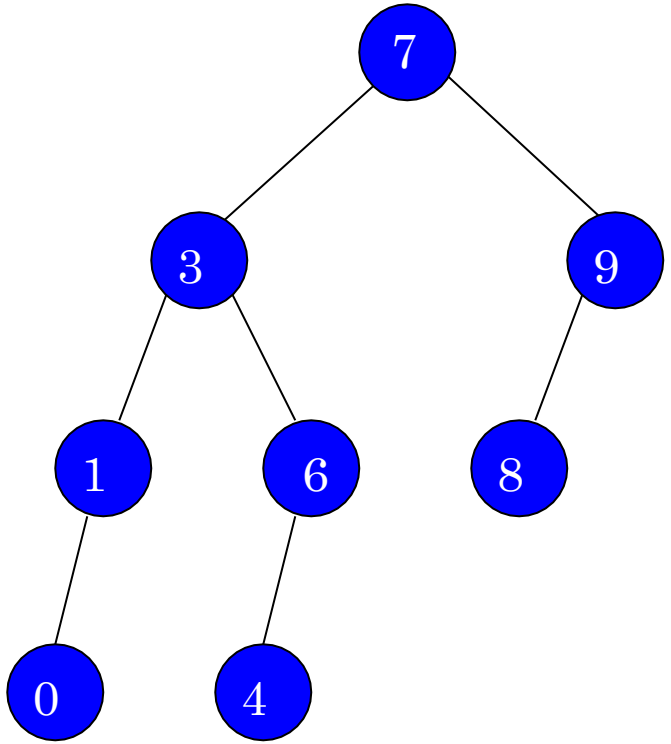
Data Structures
Binary Search Trees

COMP128 Data Structures

Ordering a Tree

Suppose that a tree contains node values that are Comparable.

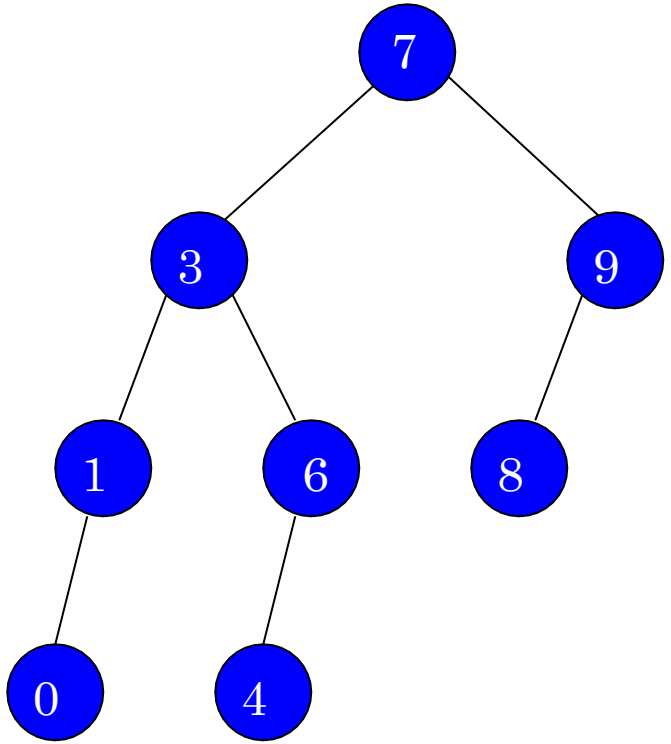
This would permit ordering node content according to < and >. The best known of all such trees is called a Binary Search Tree.



Binary Search Tree

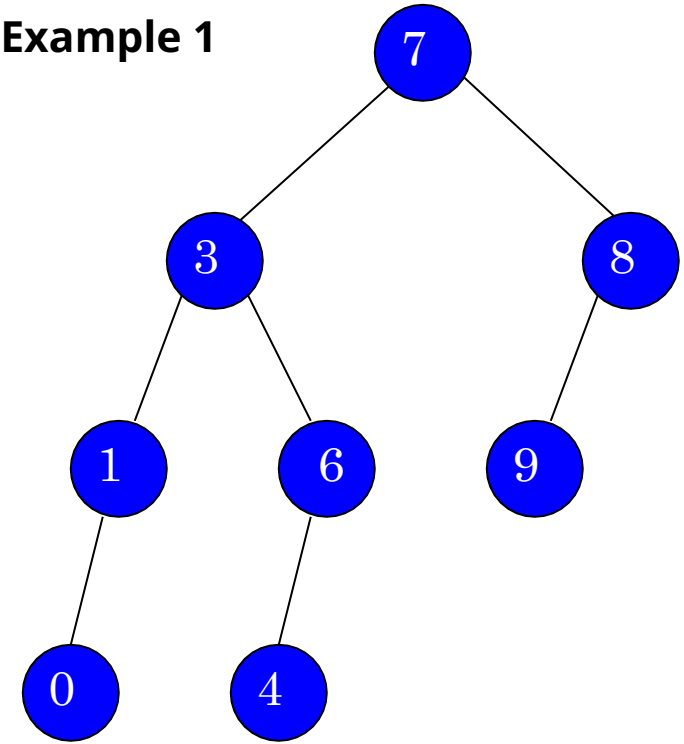
A binary search tree has two properties:

- It is a binary tree of nodes that are Comparable.
- For every node the content of its left subtree is less than the node's own content and the right subtree is greater.

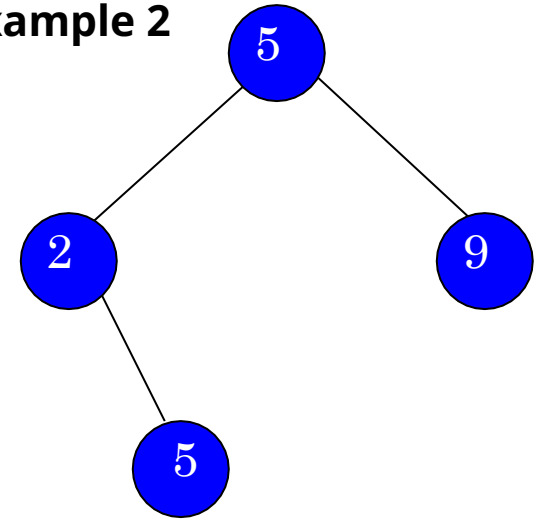


Why aren’t these binary search trees?

Example 1

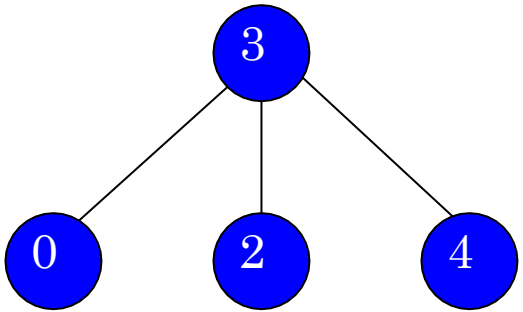


Example 2

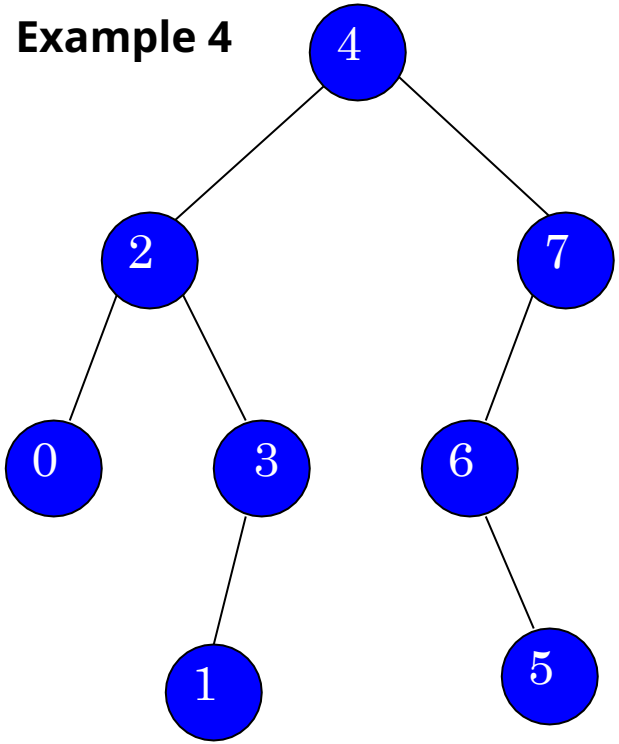


Why aren’t these binary search trees?

Example 3



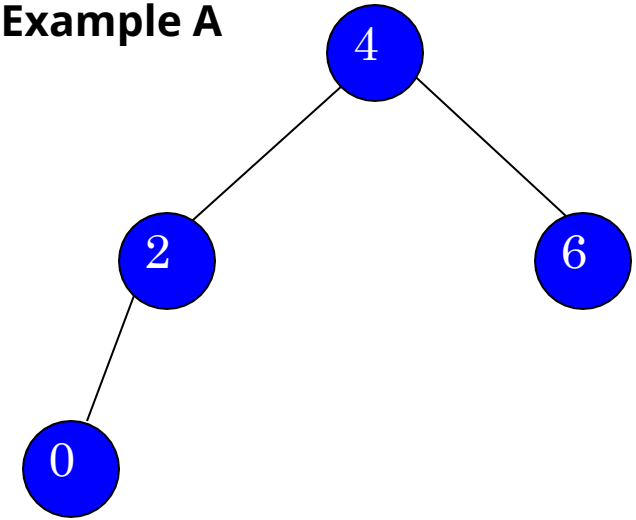
Example 4



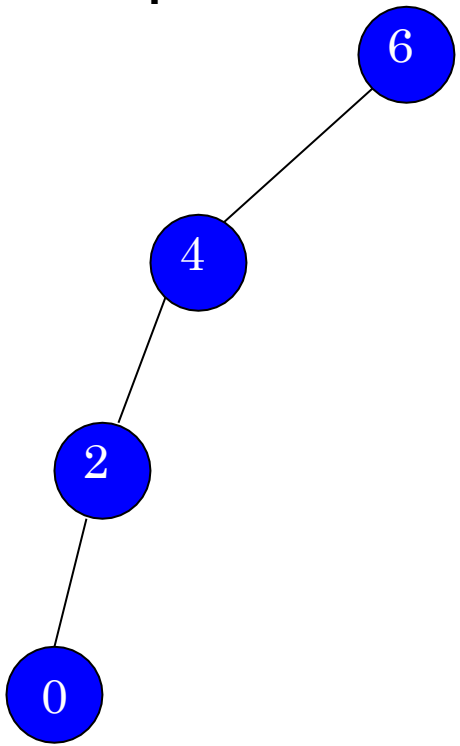
Different Representations

Note that a single data set can be represented by multiple BSTs.

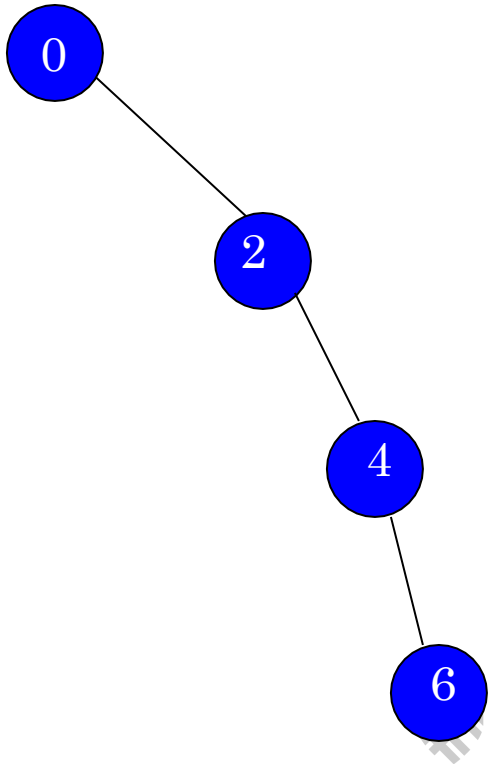
Example A



Example B



Example C



Inserting into a BST

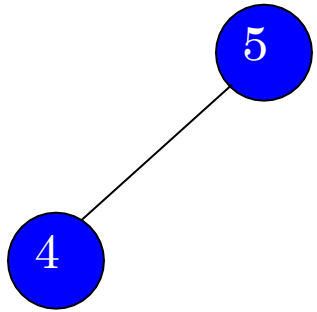
The algorithm for inserting a value within a BST follows the pattern of the BST search.

Example: Begin by inserting 5 into an empty tree.

Inserting into a BST

The algorithm for inserting a value within a BST follows the pattern of the BST search.

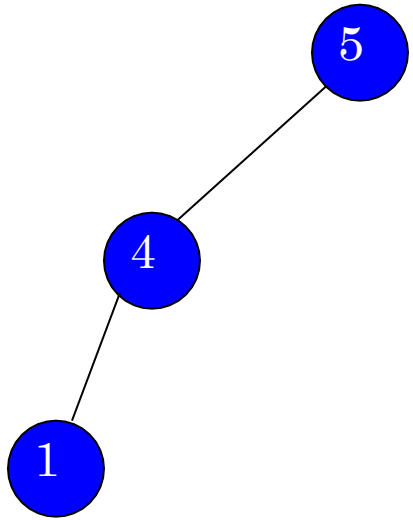
Example: Next insert 4.



Inserting into a BST

The algorithm for inserting a value within a BST follows the pattern of the BST search.

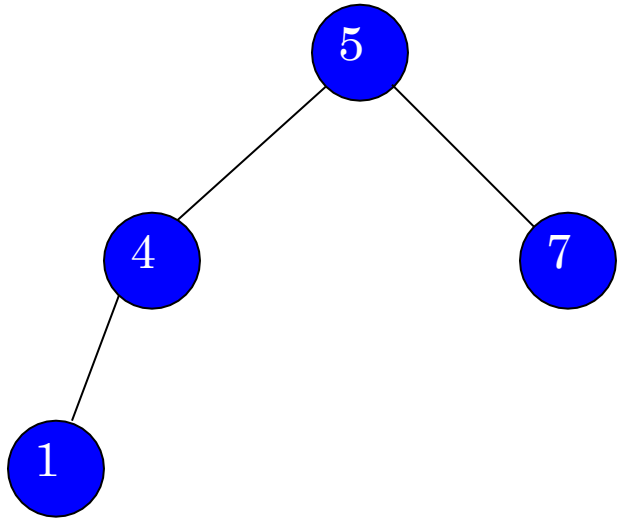
Example: Next insert 1.



Inserting into a BST

The algorithm for inserting a value within a BST follows the pattern of the BST search.

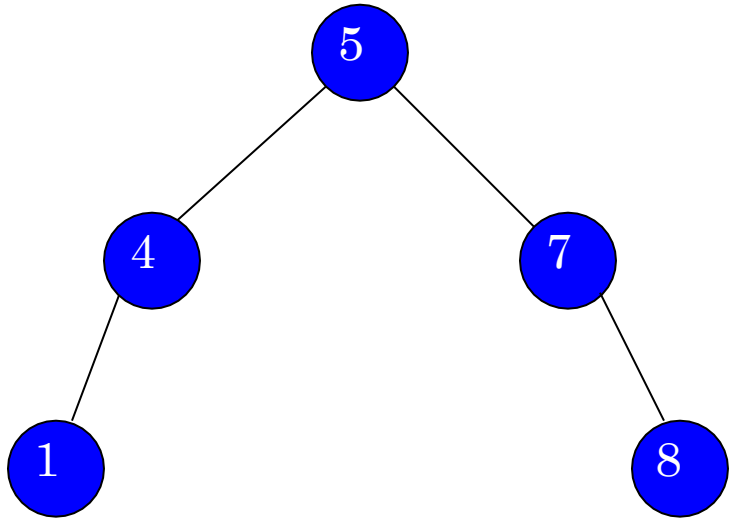
Example: Next insert 7.



Inserting into a BST

The algorithm for inserting a value within a BST follows the pattern of the BST search.

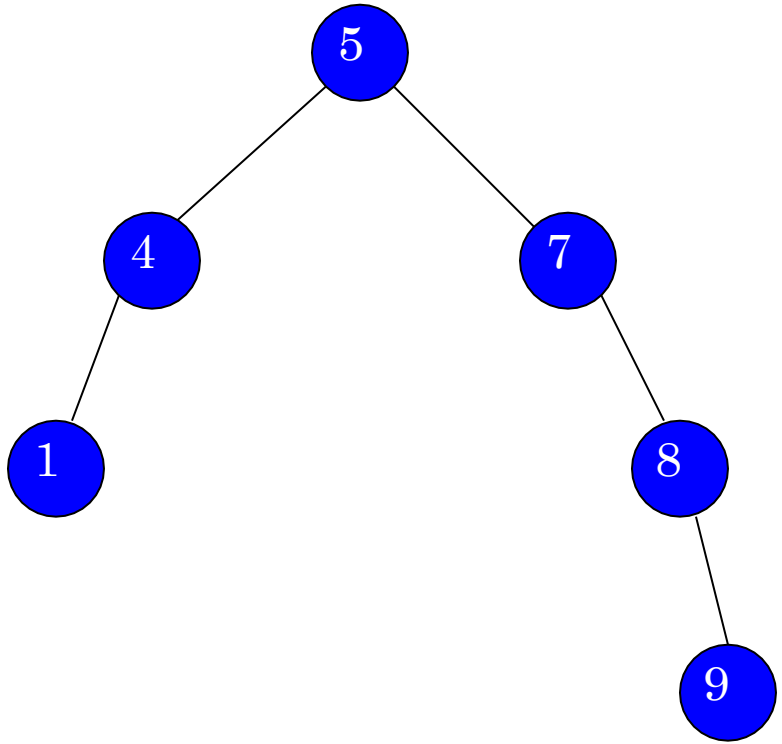
Example: Next insert 8.



Inserting into a BST

The algorithm for inserting a value within a BST follows the pattern of the BST search.

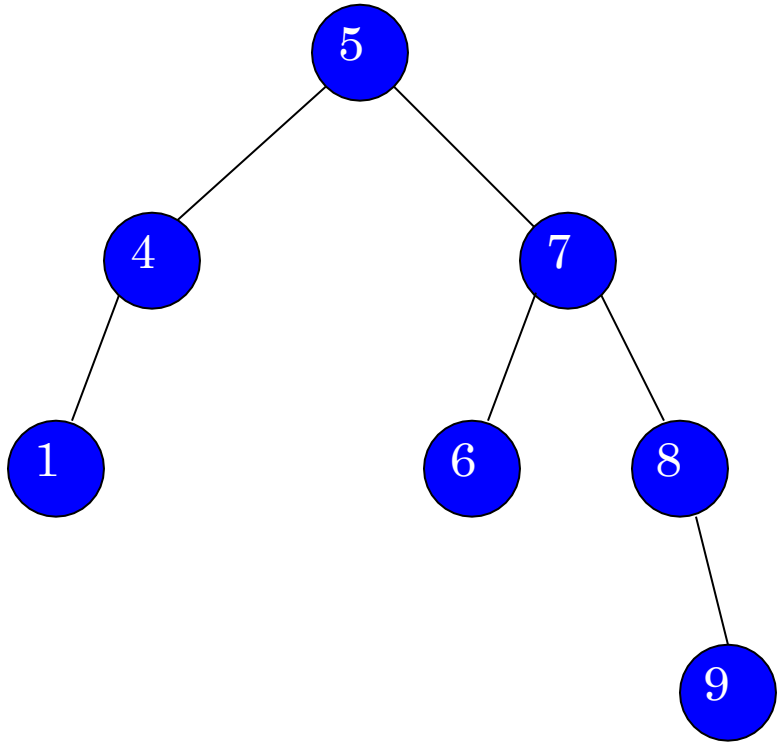
Example: Next insert 9.



Inserting into a BST

The algorithm for inserting a value within a BST follows the pattern of the BST search.

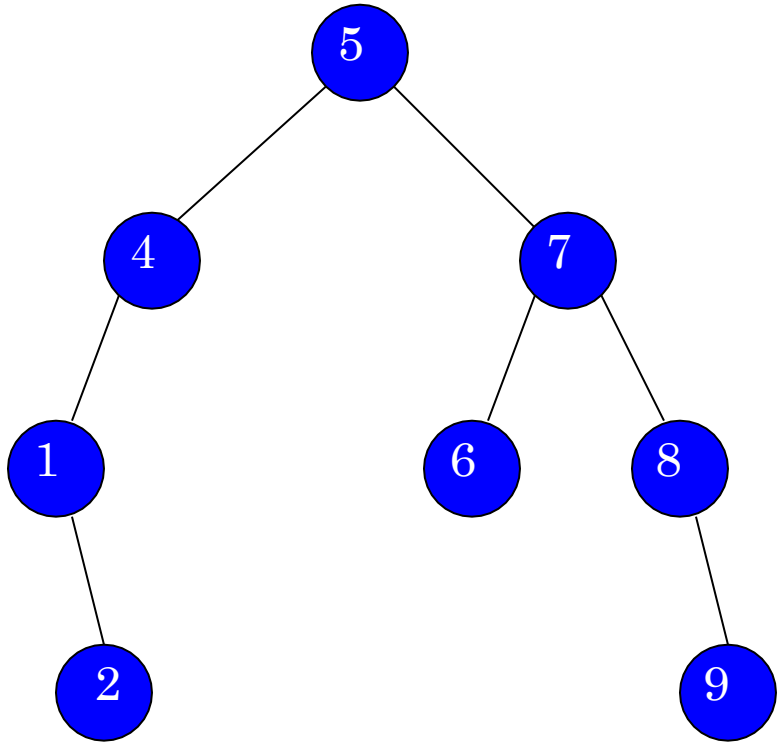
Example: Next insert 6.



Inserting into a BST

The algorithm for inserting a value within a BST follows the pattern of the BST search.

Example: Finally, insert 2.



Why use BSTs?

More Efficient Searching and Sorting

Given a binary tree of N nodes, how many must be probed to ensure that some search value is found?

- N

Given a complete BST of N nodes, how many must be probed to ensure that some search value is found?

- logN (approximately)

Why use BSTs?

More Efficient Searching and Sorting

A BST is sorted. Which of the tree traversal algorithms visits nodes in ascending order?

- In-order traversals

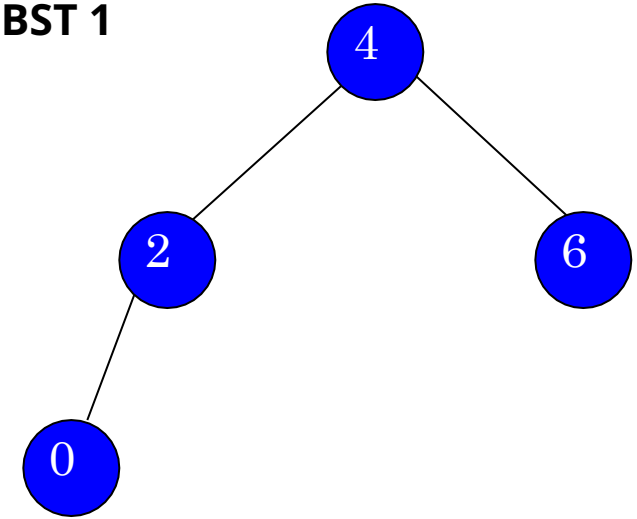
How many probes are required to sort a set of data via treesort (in the best case)?

- NlogN (approximately)

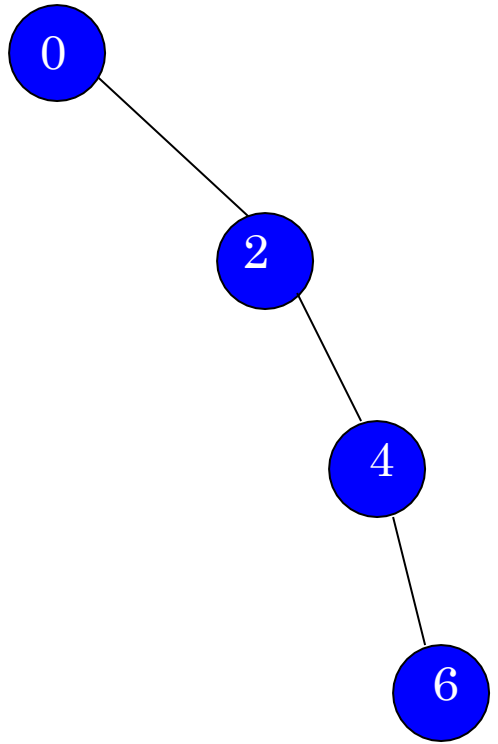
BST Performance

What is the BST shape that is least efficient to search?

BST 1



BST 2



BST Performance

Balanced Trees will be more efficient to search:

- Balance is the bushiness property of a tree.
- A complete tree is perfectly balanced.
- A tree in which no node has two children is maximally out of balance.
- An AVL tree is a BST for which every node has the following property: its left and right subtrees vary in height by at most one.
- A Red-Black tree is a balanced BST for which no two paths from the root to a leaf can differ by more than a factor of 2

In-class Activity

Binary Search Tree Activity