Name:	Jared Diehl
Course:	CS 3130
Professor:	Galina Piatnitskaia
Date:	April 18, 2020

Analysis of BST Heights

Introduction

In computer science, binary search trees (BST), sometimes called ordered or sorted binary trees, are a particular type of container: a data structure that stores "items" (such as numbers, names etc.) in memory. They allow fast lookup, addition and removal of items, and can be used to implement either dynamic sets of items, or lookup tables that allow finding an item by its key (e.g., finding the phone number of a person by name).

Definition

- **Search** - Check if a key exists in the tree.
- **Insert** - Insert a new node with the key value or increment the node's count that matches the key.
- **Delete** - Delete the node matching the key or decrement the node's count that matches the key.
- **Height** - Get the height of the tree.
- **Smallest** - Get the smallest key value in the tree.
- **Largest** - Get the largest key value in the tree.
- **Preorder Traversal (nlr)**
- **Inorder Traversal (lnr)**
- **Postorder Traversal (lrn)**

Objective

Find the average BST heights for the types of arrays containing:
- 100 integers: BST groups of 5, 10, 15.
- 500 integers: BST groups of 5, 10, 15.
- 1000 integers: BST groups of 5, 10, 15.

Computer Specifications

OS Name: Microsoft Windows 10 Home

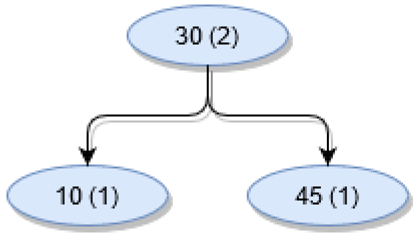
System Manufacturer: TOSHIBA
System Model Satellite: P855
System Type: x64-based PC
Processor: Intel Core i5-3210M CPU @ 2.50GHz, 2501 Mhz, 2 Cores, 4 Logical Processors
Installed Physical Memory (RAM): 8.00 GB

Programming Language Specifications

Java 1.8.0_241
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)

Handling Duplicates

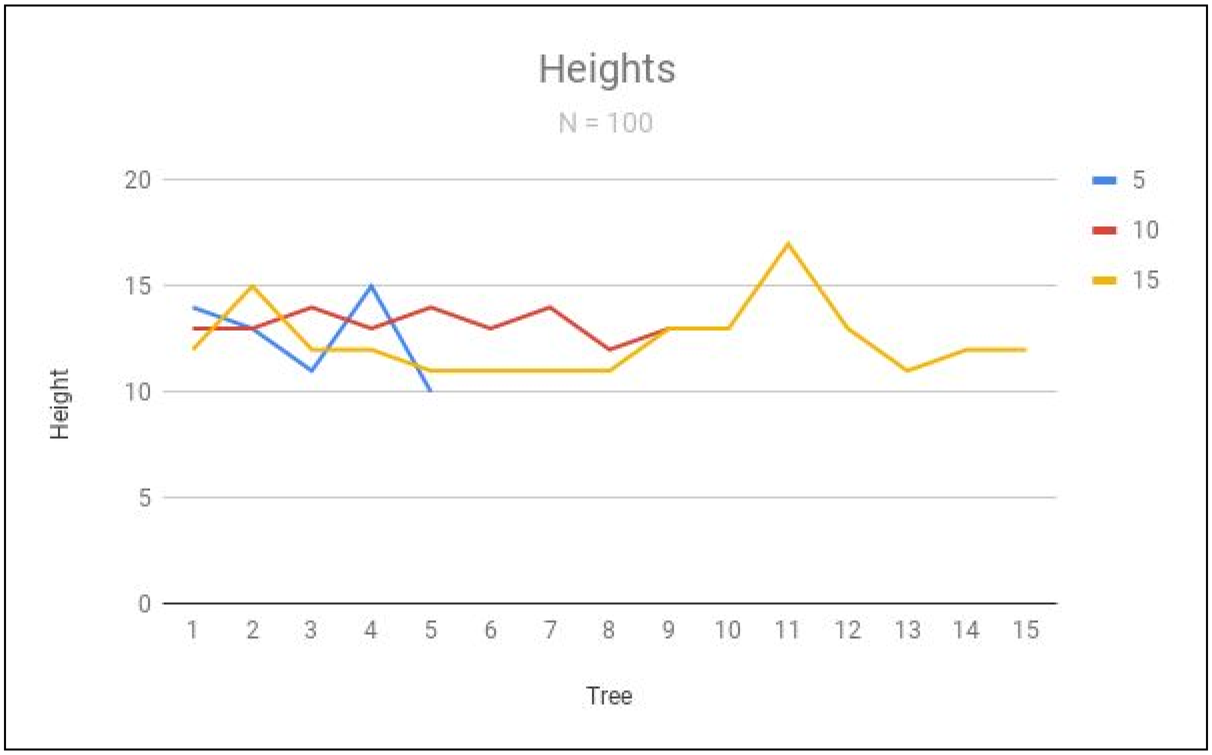
Duplicate elements are kept. A solution is to augment a tree node to store a count together with the key. Insertion of keys 30, 10, 45, and 30 in an empty BST would create the following:

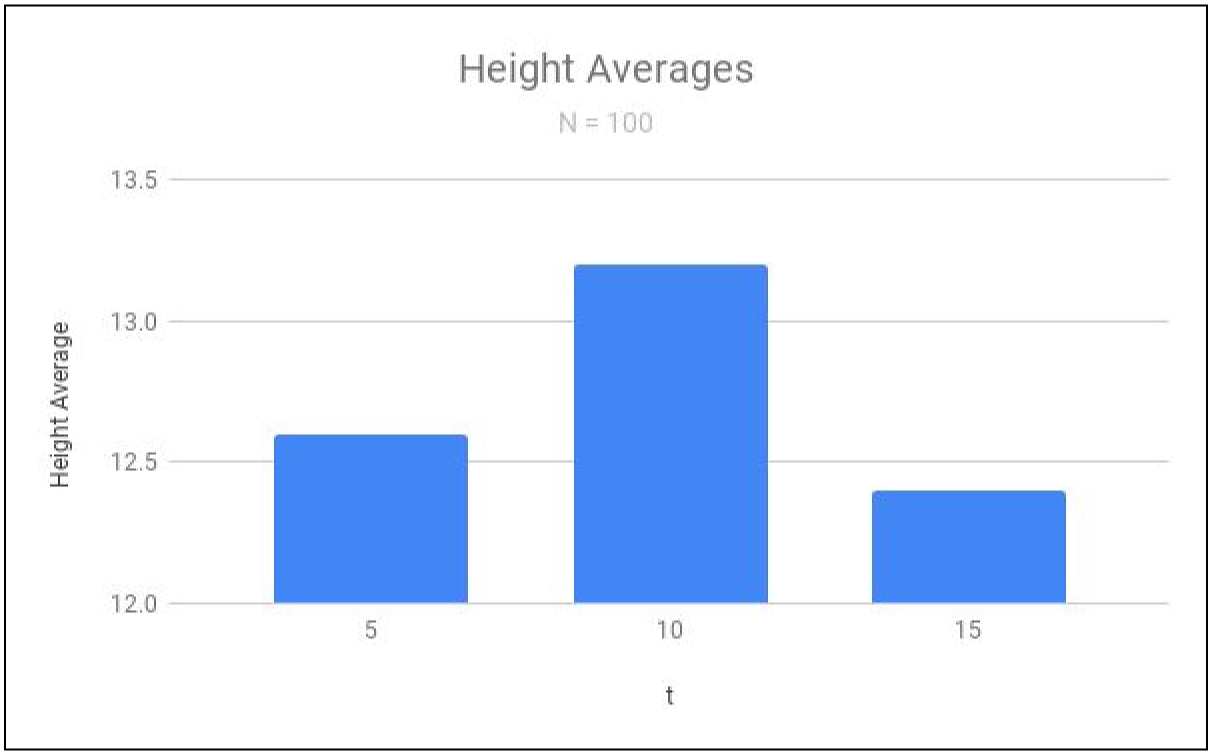


Experimental Data

BST (N = 100)			
Tree	**5**	**10**	**15**
1	14	13	12
2	13	13	15
3	11	14	12
4	15	13	12
5	10	14	11
6		13	11

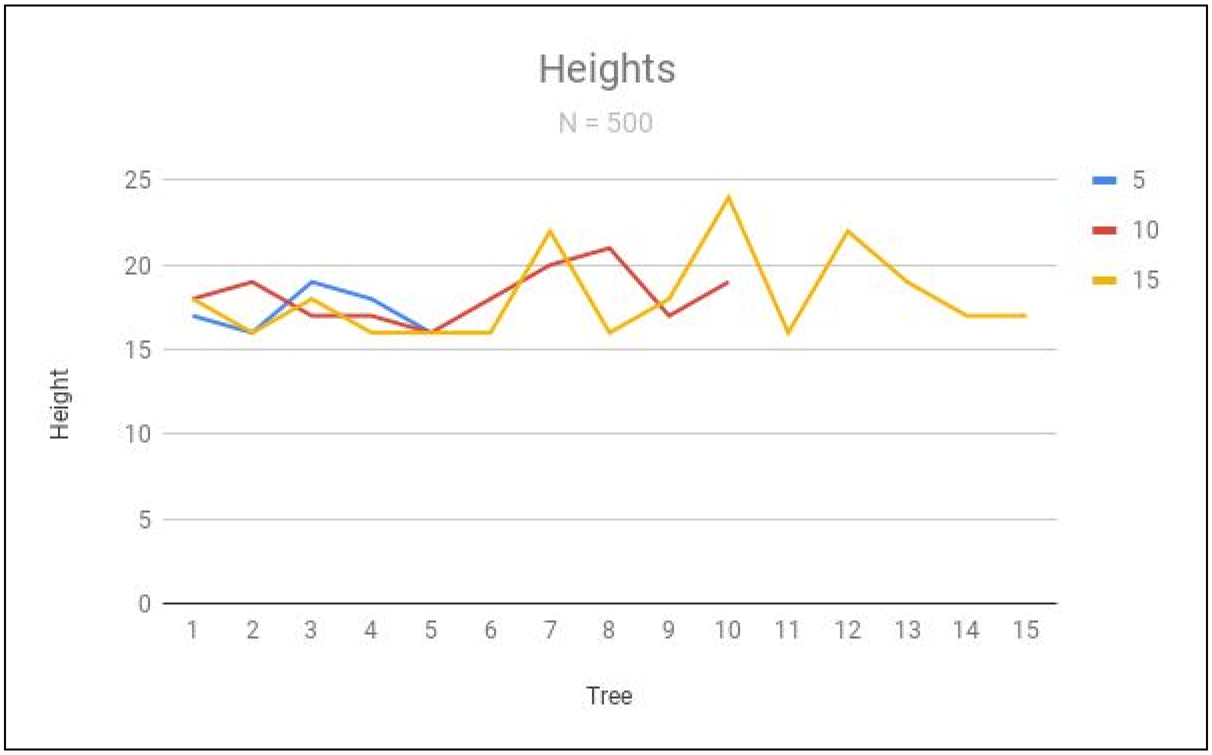
7		14	11
8		12	11
9		13	13
10		13	13
11			17
12			13
12			11
13			12
14			12
15			12
Average	12.6	13.2	12.4

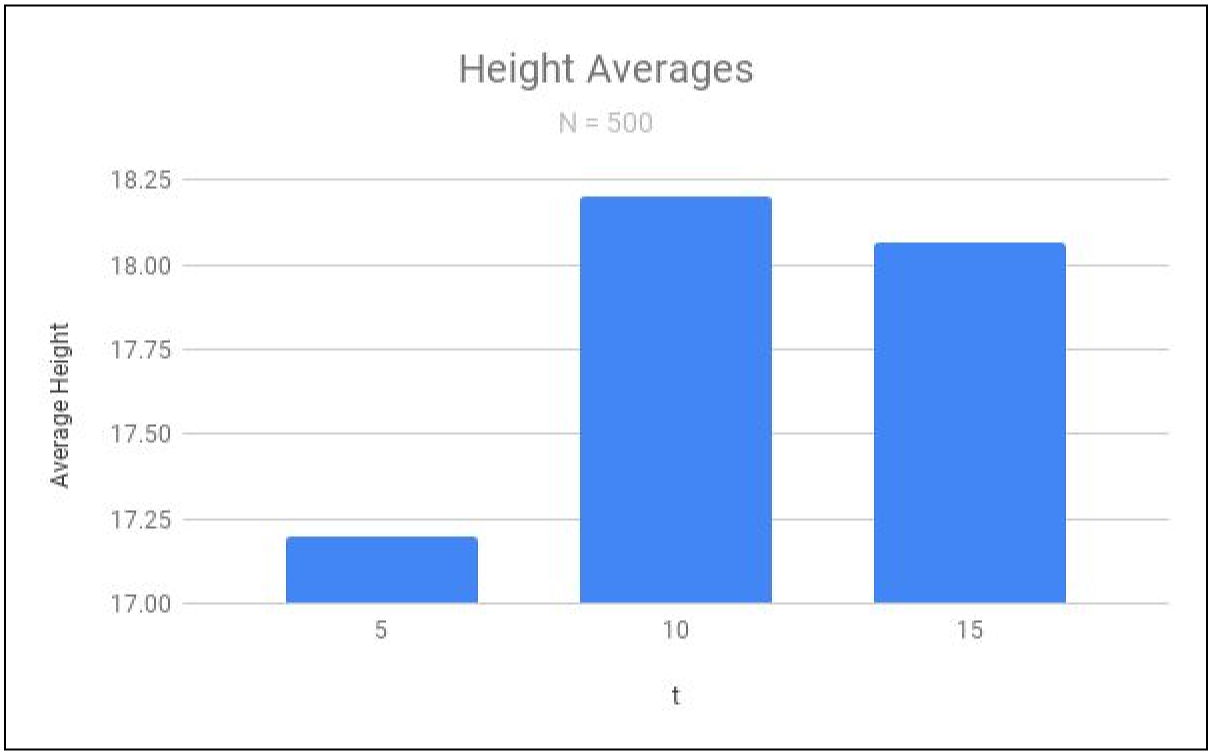




BST (N = 500)			
Tree	**5**	**10**	**15**
1	17	18	18
2	16	19	16
3	19	17	18
4	18	17	16
5	16	16	16
6		18	16
7		20	22
8		21	16
9		17	18
10		19	24
11			16

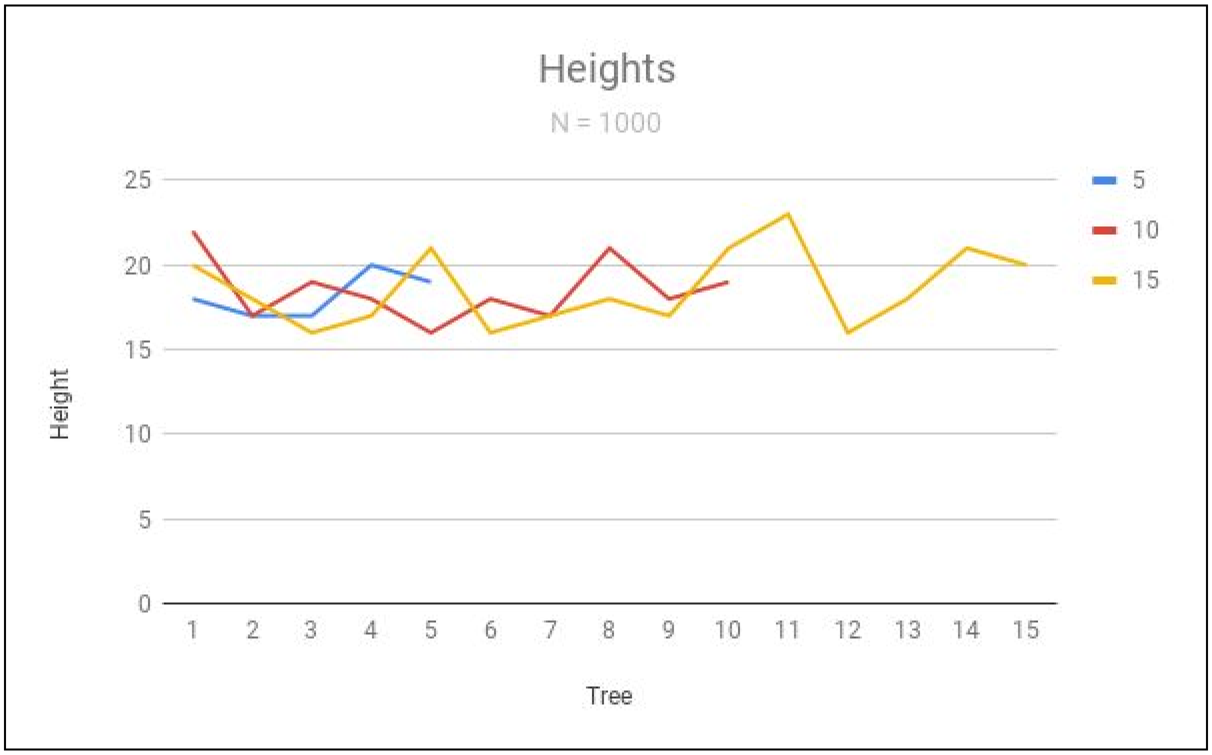
12			22
12			19
13			17
14			17
15			18
Average	17.2	18.2	18.1

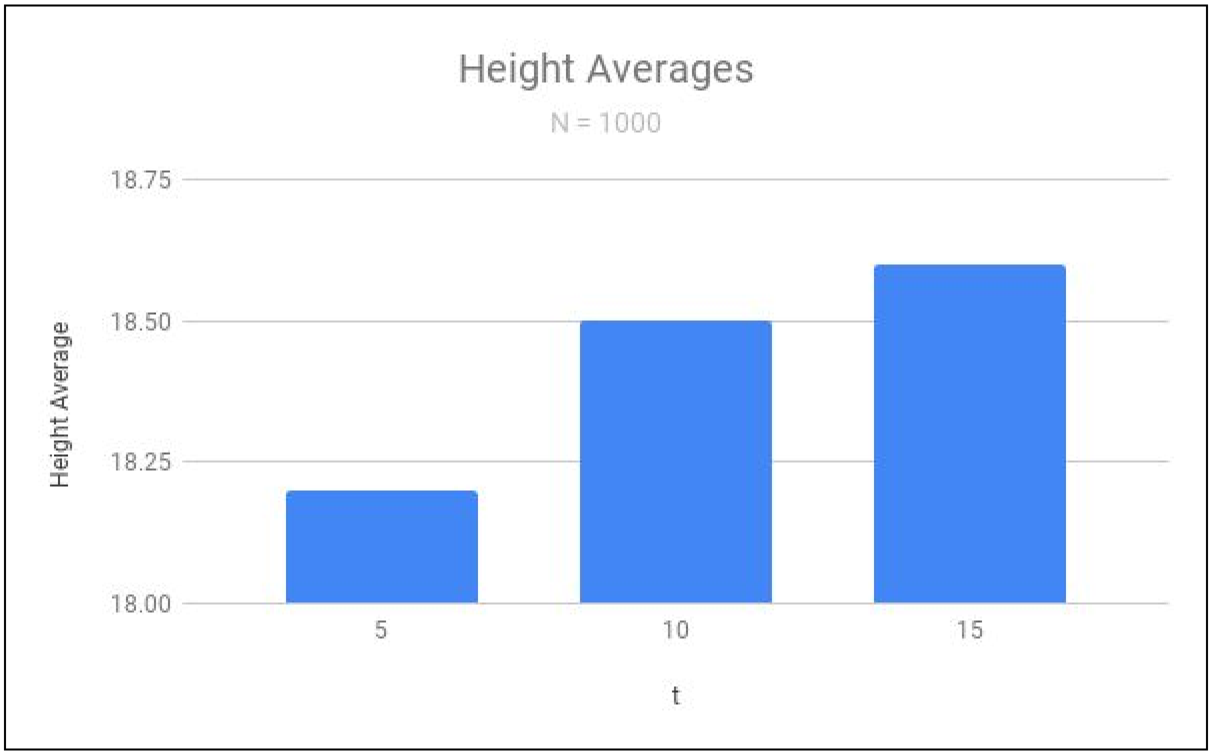




BST (N = 1000)			
Tree	**5**	**10**	**15**
1	18	22	20
2	17	17	18
3	17	19	16
4	20	18	17
5	19	16	21
6		18	16
7		17	17
8		21	18
9		18	17
10		19	21
11			23

12			16
12			18
13			21
14			20
15			20
Average	18.2	18.5	18.6





Conclusion

Based on the experimental data, the average BST heights are:

- N = 100, t = 5: 12.6
- N = 100, t = 10: 13.2
- N = 100, t = 15: 12.4
- N = 500, t = 5: 17.2
- N = 500, t = 10: 18.2
- N = 500, t = 15: 18.1
- N = 1000, t = 5: 18.2
- N = 1000, t = 10: 18.5
- N = 1000, t = 15: 18.6

The time efficiency of getting a BST's height is O(n).