# Programming Assignment 4 Report

Kevin Chou

November 2, 2015

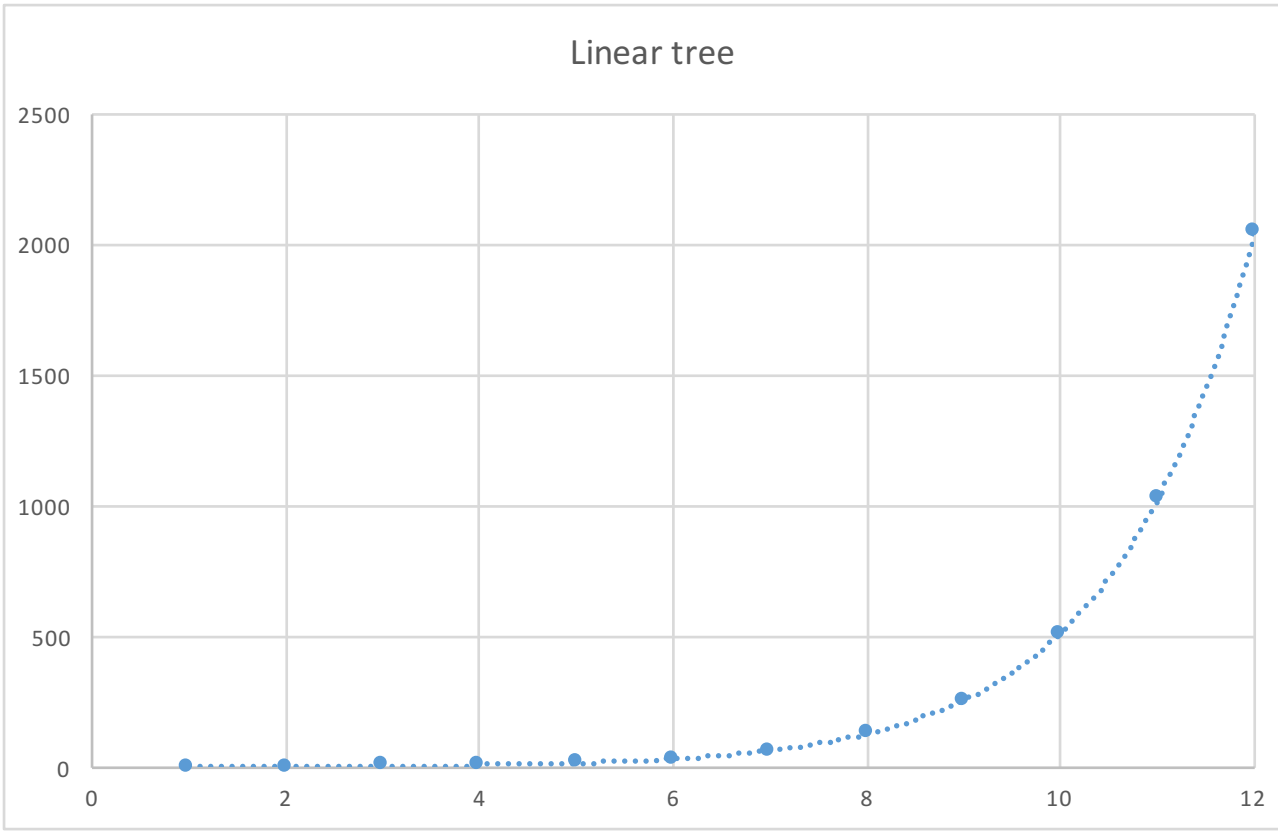
The objective of this assignment is to understand binary trees. My program contains 4 files: BinaryNode.h, BinaryNode.cpp, BinaryTree.h, BinaryTree.cpp, and main.cpp. Two classes were created: a BinaryTree class and a BinaryNode class. The BinaryTree class contains functions that define its height, size, merge, insert and remove nodes, merge. The BinaryNode class defines the search cost and left and right.

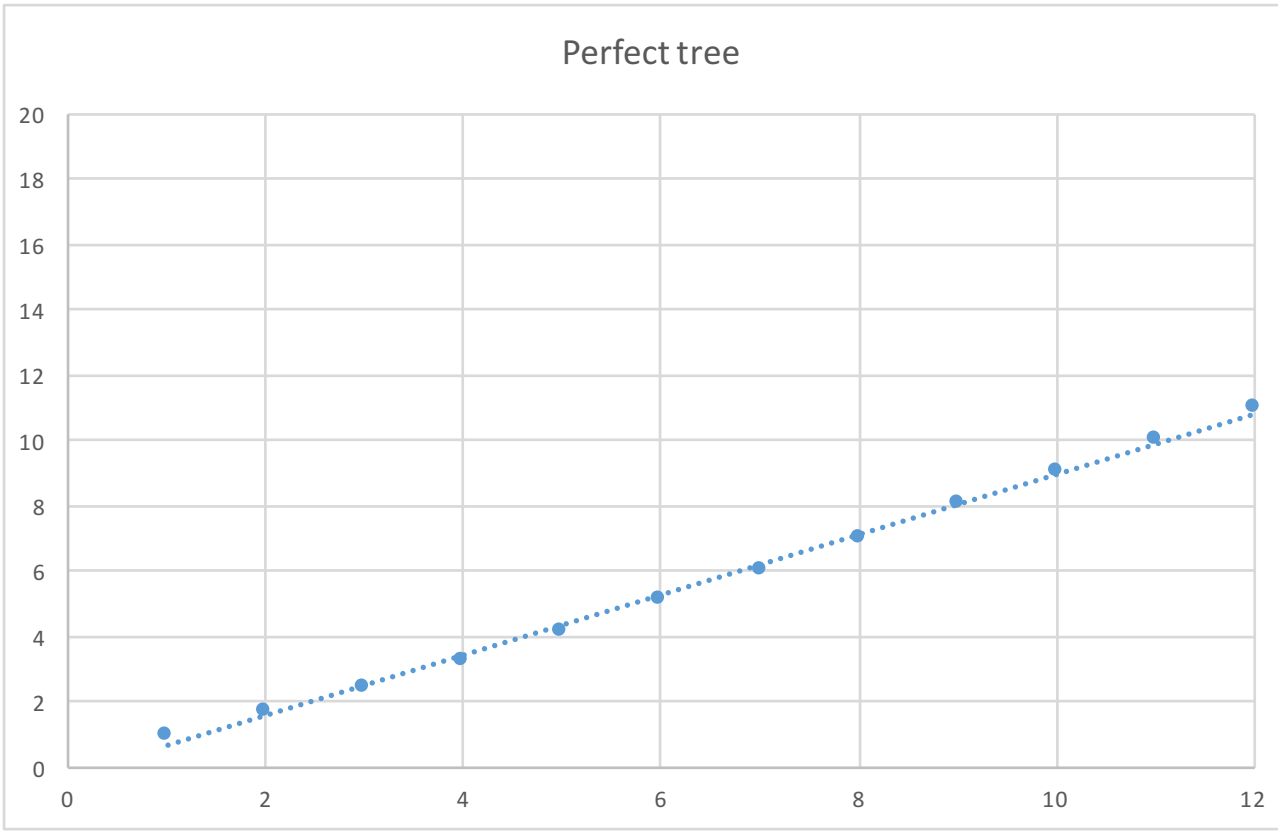
The individual search cost is calculated using the number of comparisons each time an element is inserted into the tree. The average search cost is calculated by dividing the sum of the all the search costs of the nodes in the tree by the tree's size. The time complexity is $O(\log_2 n)$. The formulas given is the total search cost in the binary tree and if we divide those equations by n we get $O(\log_2 n)$ and $O(n)$. $O(\log_2 n)$ time complexity is for a perfect tree and $O(n)$ time complexity is for a linear tree.

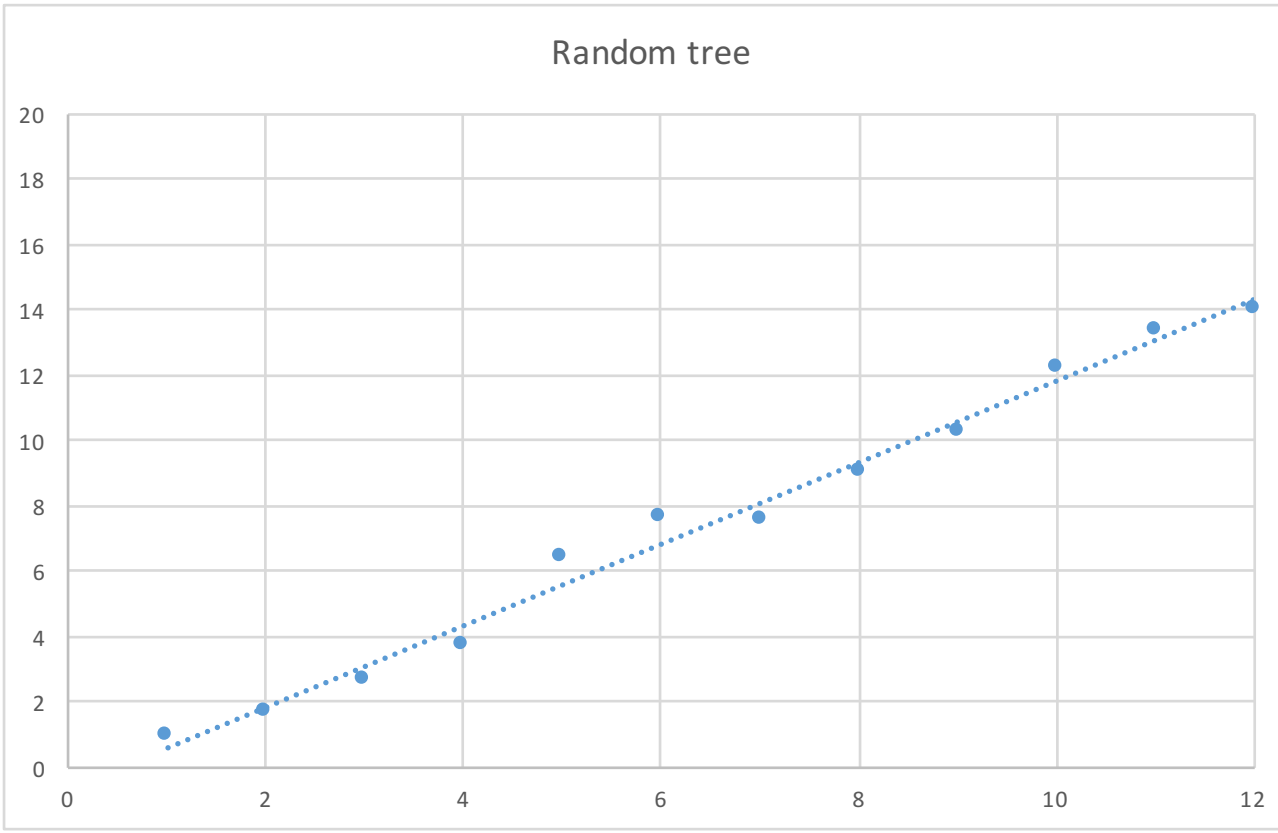
## Table for average search costs

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Linear tree (l) | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 |
| Perfect tree (p) | 1 | 1.67 | 2.42 | 3.27 | 4.17 | 5.10 | 6.06 | 7.03 | 8.02 | 9.01 | 10.01 | 11.01 |
| Random tree (t) | 1 | 1.67 | 2.71 | 3.73 | 6.39 | 7.67 | 7.59 | 9.07 | 10.30 | 12.25 | 13.40 | 14.02 |

## Graph for average search costs




Perfect tree
20
18
16
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12


Random tree
20
18
16
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12