

CSCE 221 Assignment 4 Cover Page

First Name Hunter Last Name Cleary UIN 625001547

User Name hncleary E-mail address hncleary@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more on Aggie Honor System Office website: <http://aggiehonor.tamu.edu/>

Type of sources				
People				
Web pages (provide URL)	Listed Below			
Printed material	Data Structures and Algorithms (Textbok)			
Other Sources				

<https://www.geeksforgeeks.org/get-level-of-a-node-in-a-binary-tree/>
<https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/>
<https://www.cs.usfca.edu/~galles/visualization/BST.html>
http://www.algolist.net/Data_structures/Binary_search_tree/Insertion
<http://www.cplusplus.com/forum/general/31509/>
<http://code.activestate.com/recipes/577552-binary-search-tree/>
<http://www.cplusplus.com/forum/beginner/75440/>
<http://www.cplusplus.com/reference/queue/queue/>

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.
On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.

Your Name Hunter Cleary

Date 4-1-2018

Assignment 4 (100 pts)

Program: Due April 2 at 11:59 pm

Objectives: Programming (70 points)

Report (30 points)

Write a brief report that includes the following:

1. A description of the assignment objective, how to compile and run your programs, and an explanation of your program structure (i.e., a description of the classes you use, the relationship between the classes, and the functions or classes in addition to those in the lecture notes).

The objective of the assignment was to create a binary search tree and then be able to calculate search costs of nodes in the structure. Two structures exist. One for the nodes within the search tree and another structure for the tree itself. The search tree contains the nodes which hold values and pointers to other nodes in the tree. The tree contains overloaded operators and functions for input and calculating search cost.

The program can be compiled using the make file.

Makefile

./main

2. A brief description of the data structure you create (i.e., a theoretical definition of the data structure and the actual data arrangement in the classes).

Binary search are a data structure that allows for faster access of data indexed by keys. The initial creation of the tree has a greater cost, but afterwards the cost of operations to find items in the tree is greatly reduced. Nodes traverse from root to leaf, typically using left and right subtrees. Each comparison in these subtrees allows the operations to skip an average of half the tree. Inserting and deleting items happens in logarithmic time.

3. A description of how you implement the calculation of (a) individual search cost and (b) average search cost and explain which tree operation (e.g. find, insert) was helpful. Analyze the time complexity of (a) calculating individual search cost and (b) **summing up** the search costs over all the nodes.

(a) Search cost calculation was implemented by creating a value for search cost in the search function of the binary search tree. ($O(n)$)

(b) To calculate the average search cost, the total cost is calculated and then divided by the number of nodes present in the tree. ($O(n)$)

4. Give individual search cost in terms of n using big-O notation. Analyze and give the average search costs of a perfect binary tree and a linear binary tree using big-O notation, assuming that the following formulas are true (n denotes the total number of integers).

$$\sum_{d=0}^{\log_2(n+1)-1} 2^d(d+1) \simeq (n+1) \cdot \log_2(n+1) - n \quad \text{and} \quad \sum_{d=1}^n d \simeq n(n+1)/2$$

Search cost of each item will be $O(\log n)$.

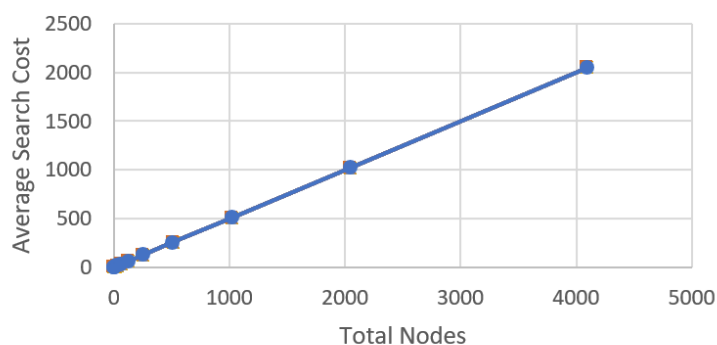
Linear Tree The height of the tree is $n-1$. Using the formula the total cost can be represented as $n(n+1)/2$. Dividing by n subproblems we get $(n+1)/2$ which is $O(n)$.

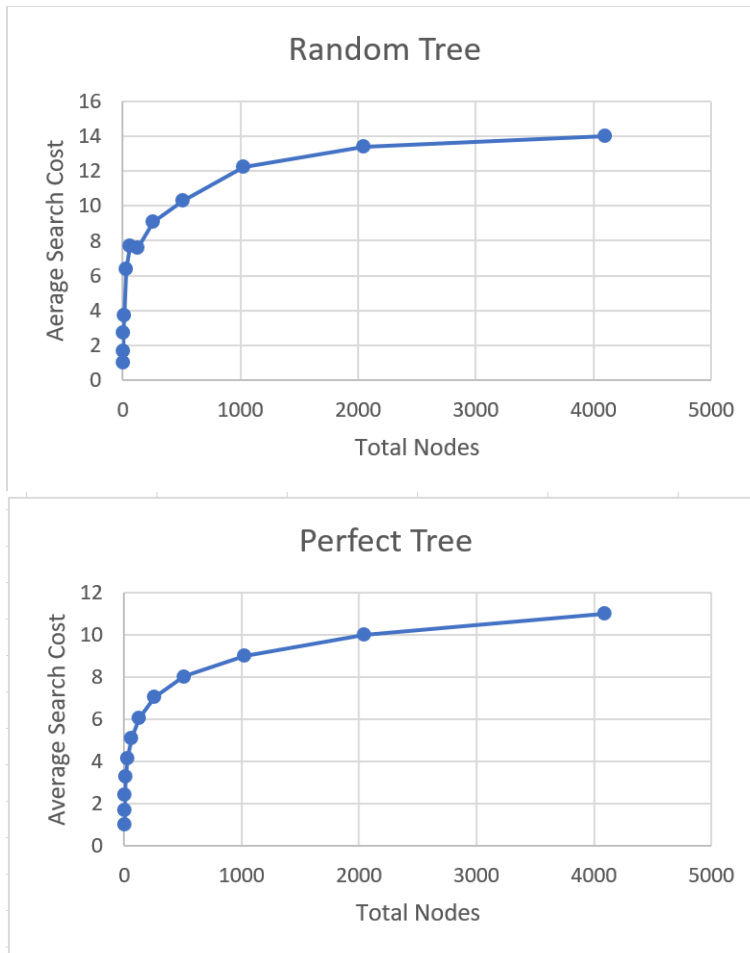
Perfect Tree The height of the tree is $\log n$. There are 2^n nodes at each level of the tree. $2^{(\log(n+1) - 1)\log(n+1)}$ which is $O(\log n)$.

5. Include a table and a plot of average search costs you obtain. In your discussions of the experimental results, compare the curves of search costs with your theoretical analysis results in item 4.

Nodes	Search Costs		
	Linear	Perfect	Random
1	1	1	1
3	2	1.67	1.67
7	4	2.43	2.71
15	8	3.27	3.73
31	16	4.16	6.39
63	32	5.1	7.67
127	64	6.06	7.59
255	128	7.03	9.07
511	256	8.02	10.3
1023	512	9.01	12.25
2047	1024	10.01	13.4
4095	2048	11	14.02

Linear Tree





The experimental results were representative of the theoretical analysis. Average / random and perfect case were both $O(\log n)$. Linear came out as $O(n)$.