CSCE 221 Assignment 4 Cover Page

First Name: Matthew Last Name: Stevens UIN: 924000693

User Name: mattizrawr E-mail address: mattizrawr@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	Joshua Langley			
Web Pages	https://piazza.com/class/ij91nvqapg47ox?cid=235			
Printed Material	none			
Other Sources	Lab T.A.'s			

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: Matthew Stevens Date: 3/22/2016

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 assignment objective is to understand how to build binary trees and to see the search-time cost of each node, including when adding and removing nodes. Compile on the Build server (not SUN) and simply just ./a.out. I separated the tree and the nodes into two separate classes and made them friends. This was so that when I was overloading in one class, I wouldn't overwrite any values that may exist in the other class. The binary node class consistently updated search cost and return the elements of the nodes. I also made a copy constructor so new nodes could easily be created. The binary tree class creates/stores the root and creates and deletes the trees when it needs to change. The functions create nodes and store values in them, as well as connects nodes by their addresses so that a certain order is followed.

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).

The binary tree data structure stores n nodes with n values and creates n-1 bridges (pointers to other nodes) where each node may not have more than 2 pointers to other nodes. In my program, a tree is created this way and when transcending and removing/adding nodes, it slows down the runtime and alters the tree.

3. A description of how you implement the calculation of (a) individual search cost and (b) average search

cost and (c) updated search cost. Is the implementation associated with the operations find, insert,

remove? Analyze the time complexity of (a) calculating individual search cost, (c) updating the

search cost of an individual node and (b) summing up the search costs over all the nodes.

Individual search cost is the search cost of getting to a single node (traversing through n-1 bridges to the nth node) and average search cost is the search cost of all the nodes divided by the number of nodes (n).

The time complexity for individual search cost is O(n) and updating the search cost of an individual node is O(n-1)+2 due to the traversal to the node and the cost of replacing a value is 2 operands as it also increments or decrements the search cost. Summing up all of the nodes is the cost of each individual node and the adding of them all together (each add is an operand) so it is O(n)+n for the sum.

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).

Individual search cost: (n terms) = O($\log_2$n). Perfect: O($\log_2$(n+1)^2)-1 linear: O((n+1)/2)

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.

Set#	Avg search cost
{8,12,14,15,13,10,11,9,4,6,7,5,2,3,1}	3.266667
{8,12,14,15,13,10,9,4,6,7,5,2,3,1}	3.2142857
{8,12,10,11,9,4,6,7,5,2,3,1}	3.25

When I used the search costs of 4, the average search cost was slightly lower than the average search costs that I have obtained in 5.

I had trouble with the attachment of the scatterplot but my results are in a parabolic shape while the theoretical formulas in 4 are linear.