CSCE 221 Cover Page
Programming Assignment #4
Due March 8 at midnight to CSNet

Jessica Fang UIN: 224003796

User Name: sndnzki E-mail address: jessicafang@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 in the Aggie Honor System Office http://aggiehonor.tamu.edu/

Type of sources					
People	Katherine Click	Devin Carr			
Web pages (provide URL)					
Printed material					
Other Sources					

I certify that I have listed all the sources that I used to develop the solutions/code 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 (signature) Jessica Fang Date 04/06/2015

1) This program is able to take in numbers from an input file, create a binary search tree, and perform elementary operations such as insert and remove on the tree. The purpose of this assignment is to utilize our knowledge of data structures and binary search trees to create a program which prints out binary search trees and calculate the search search cost of the elements. An "Empty" exception is thrown when functions try to call any find or remove functins on an empty tree. This program is compiled using a makefile (type make) and run using ./main.

2)The program consists of a binary tree class which is implemented using nodes that hold data and pointers to a left and a right child. The class has a few simple functions such as inserting and removing nodes and printing out the binary tree level by level as well as preorder, inorder, and postorder.

3) How search costs are calculated:

a) individual search cost. calculated when the data is added into the binary tree.

b) average search cost. calculated by summing up the individual search costs of each node and then dividing by the tree's size.

c) updated search cost. search costs are updated recursively by traversing through the levels and using a counter.

Time Complexity of

a) individual search cost. best: O(log n) avg: O(log n) worst: O(n)

b) average search cost. best: O(n log n) avg: O(n log n) worst: O(n^2)

c) updated search cost. best: O(n log n) avg: O(n log n) worst: O(n^2)

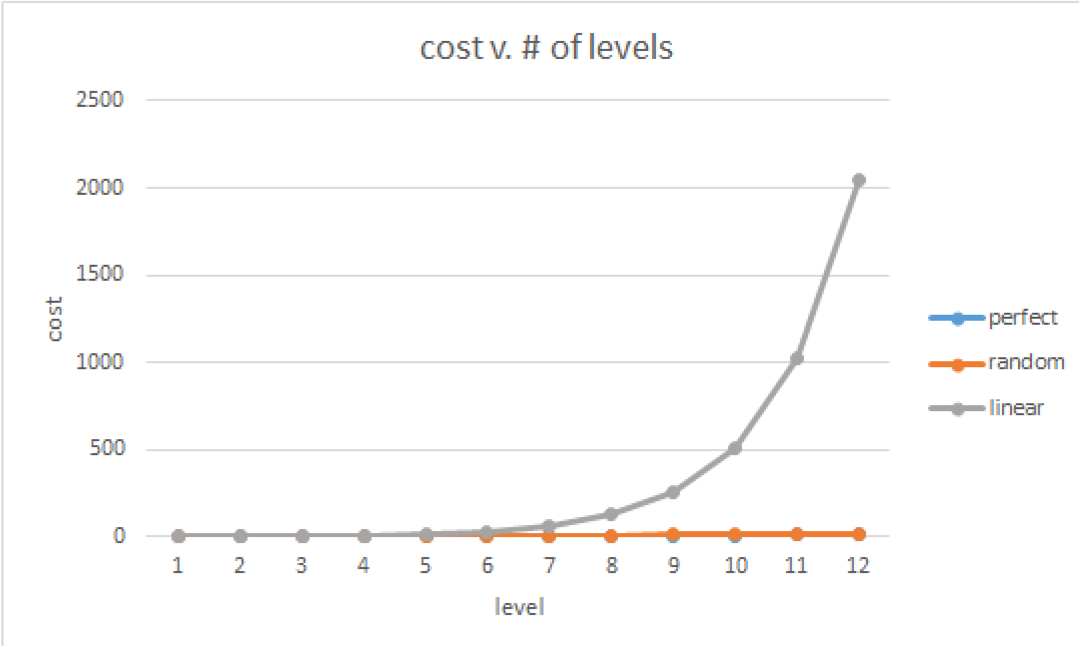
4) Using the given formulas, one can derive the individual search costs of perfect and linear trees as O(log n) and O(n) (respectively) by dividing by n.

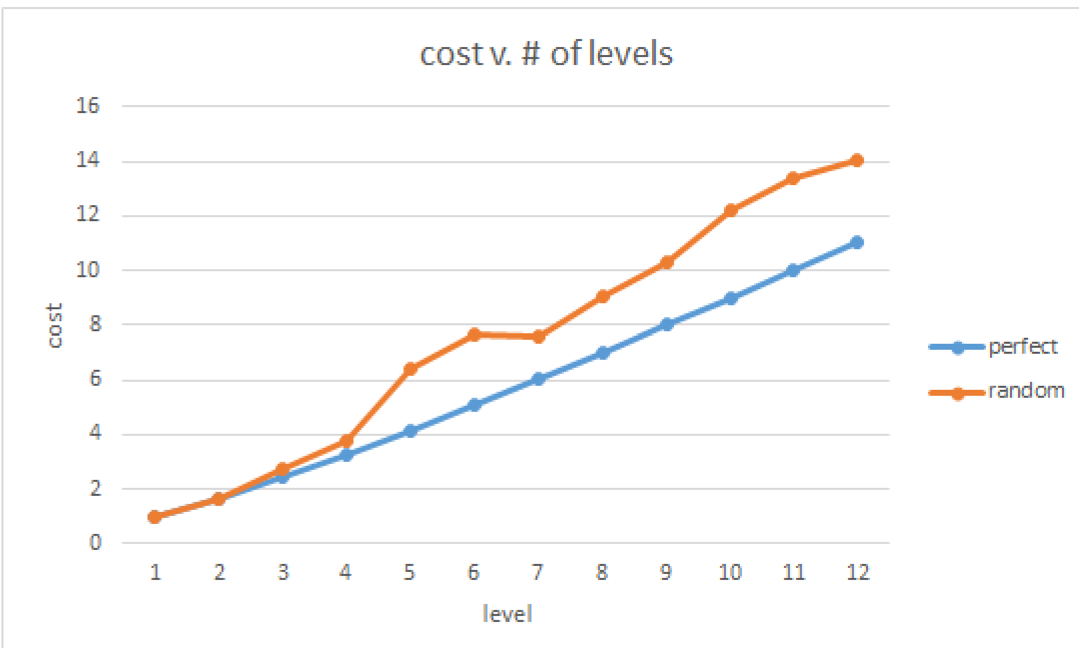
5) Cost:

1p	2p	3p	4p	5p	6p	7p	8p	9p	10p	11p	12p
1	1.66667	2.42857	3.26667	4.16129	5.09524	6.05512	7.03137	8.01761	9.00978	10.0054	11.0029

1r	2r	3r	4r	5r	6r	7r	8r	9r	10r	11r	12r
1	1.66667	2.71429	3.73333	6.3871	7.66667	7.59055	9.06667	10.3033	12.2463	13.3972	14.0237

1l	2l	3l	4l	5l	6l	7l	8l	9l	10l	11l	12l
1	2	4	8	16	32	64	128	256	512	1024	2048


cost v. # of levels
2500
2000
1500
1000
500
0
cost
1
2
3
4
5
6
7
8
9
10
11
12
level
perfect
random
linear


cost v. # of levels
16
14
12
10
8
6
4
2
0
cost
1
2
3
4
5
6
7
8
9
10
11
12
level
perfect
random