Report
Programming Assignment #4

First Name: Pratik Patel UIN: 527004337
User Name: p.pratik99 E-mail address: p.pratik99@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.

Type of sources	1	2	3
People			
Web pages (provide URL)			
Printed material	Textbook		
Other Sources	Lecture Slides		

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).

   a. Objective: The main objective of this assignment is to implement a Binary Search Tree data structure. It also asks for implementation of various function of Binary Search Tree class such as insert, find, print etc. It also asks for a comparison to other data structure in terms of Big-Oh notation.
   b. To compile this assignment, just run the following command on any Unix based system, g++ -std=c++11 main.cpp BTree.cpp. In order to run the program on the command line, execute the following command, ./a.out.
   c. There are main three files, BTree.h, BTree.cpp and main.cpp. BTree.h contains the declaration of struct Node as well as the declaration of the class BTree. BTree.cpp contains definitions to the functions declared in BTree.h. main.cpp is a driver file. It interacts between the user and BTree class.

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

   I have created a Binary Search Tree. I have also used a queue to implement one functionality of the Binary Search Tree. In binary search tree each Node has two pointers, a value and an integer specifying its search cost in the tree. In Binary Search Tree, each Node to the left of a given Node has to be less than the given Node and each Node on the right is greater than the given Node.

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. I calculated each search cost at the time of insertion. I have added the number of comparisons it took to find its place in the tree plus 1, which gives me the search cost of that Node. Therefore, the time complexity of different cases are as follows. $Average: O(log_2 n), Best: O(1), Worst: O(n)$

   b. To calculate the Average search cost, I have first calculated the total search cost by calling a function, which adds the search cost of the current node to the final result, on every Node in the tree recursively. Then, I divide that result by the size of the tree given by the get_size() function. Therefore, the time complexity is $O(n)$ for any given input sequence.

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

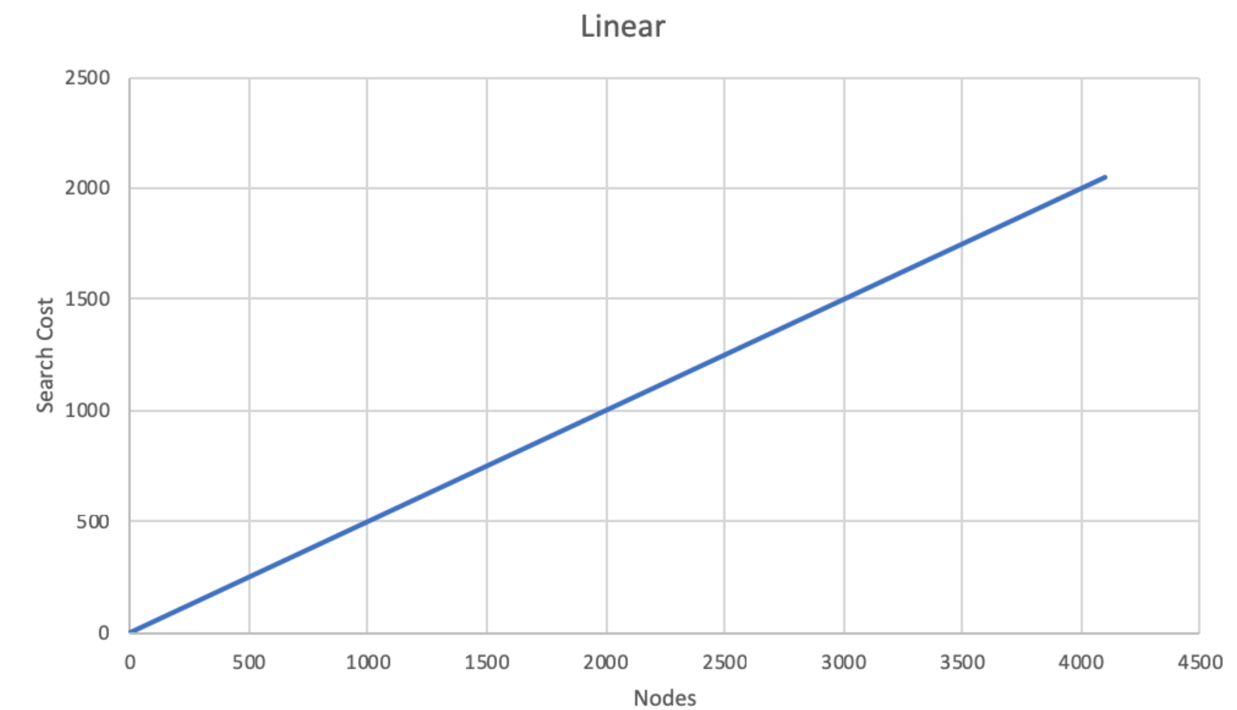
notation, assuming that the following formulas are true (n denotes the total number of nodes).

a. Linear Tree: We know that the height of the tree is $n - 1$, using the given formula the total cost becomes $\frac{n(n+1)}{2}$. Therefore, the average cost will be $\frac{\frac{n(n+1)}{2}}{n} = \frac{n+1}{2}$. Therefore, the Big-O notation for the average cost is $O(n)$.

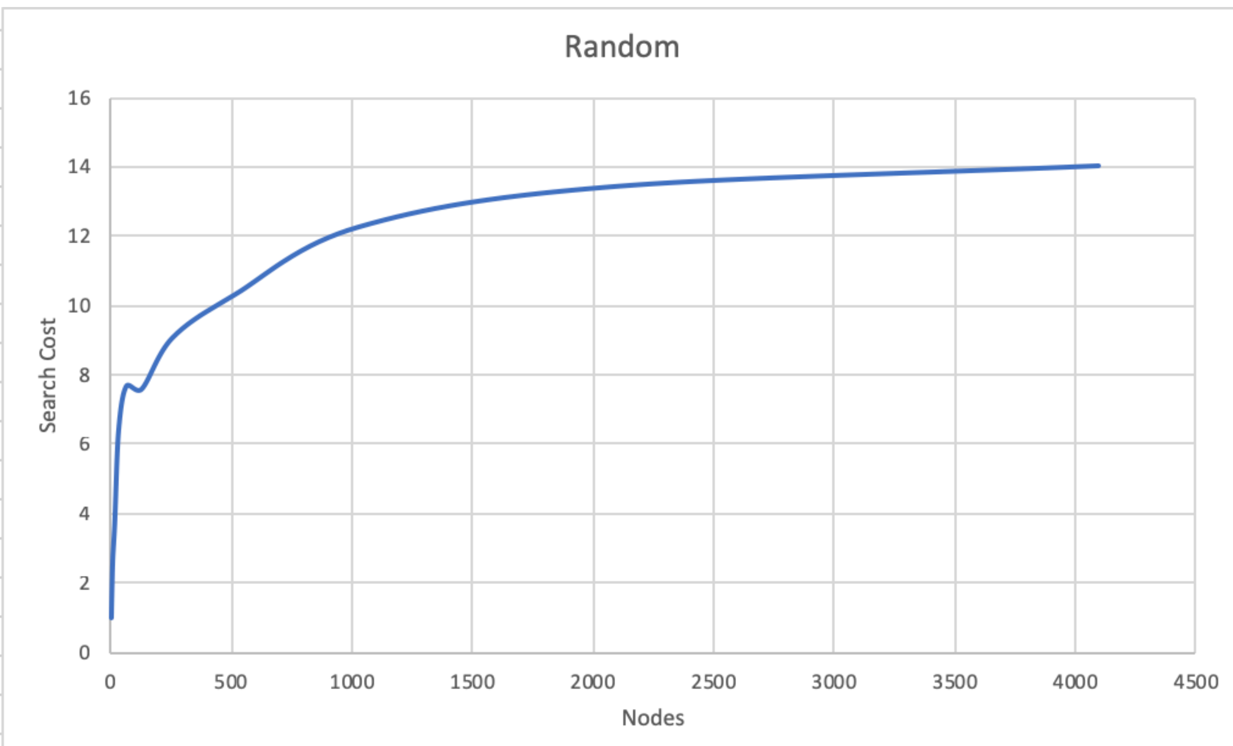
b. Perfect Tree: the height a perfect tree is $log_2 n$. Therefore, there are 2^n nodes on each level. Using the given formula the search cost will be $log_2(1 + 1)\ +$ $2(log_2(2 + 1))$ + $2^2 log_2(3 + 1) + \cdots + 2^{log_2(n+1)-1} log_2(n + 1)$, which is equal to $(n + 1)(log_2(n + 1) - n)$. Therefore, the Big-O notation for the average cost is $O(log_2 n)$.

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

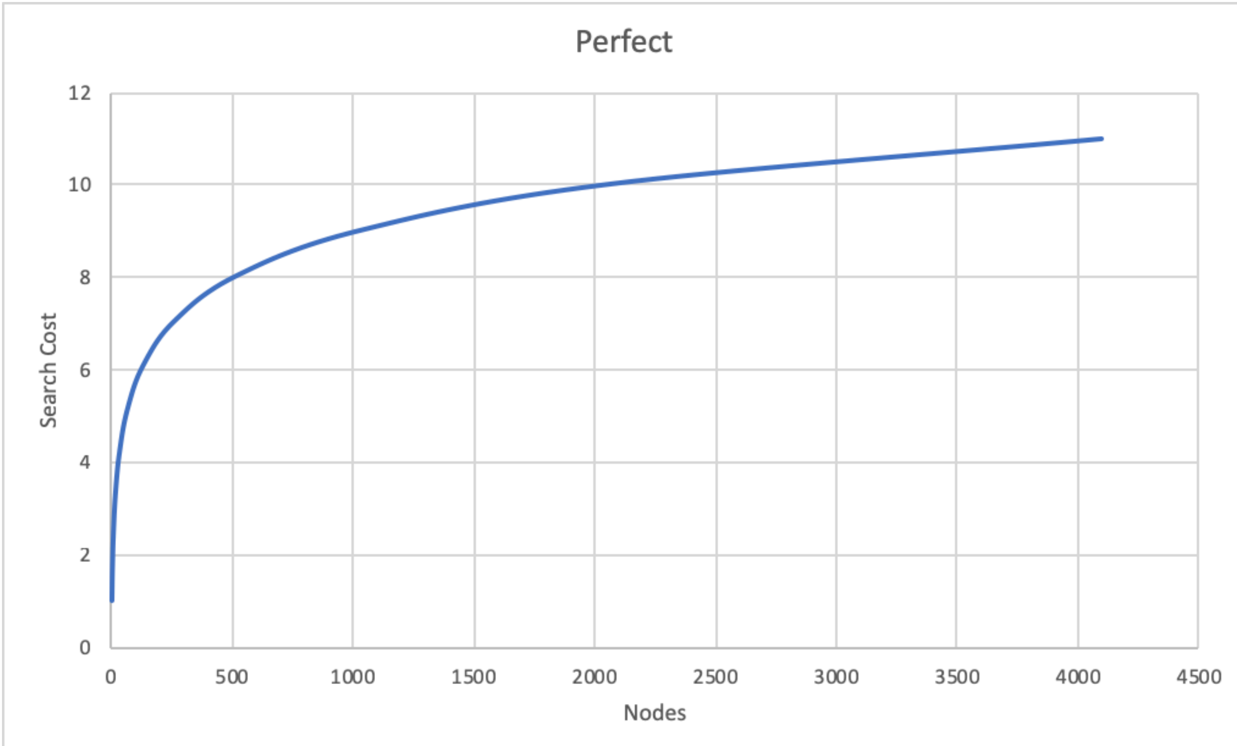
Nodes	**Linear**	**Random**	**Perfect**
1	1	1	1
3	2	1.67	1.67
7	4	2.71	2.43
15	8	3.73	3.27
31	16	6.39	4.16
63	32	7.67	5.1
127	64	7.59	6.06
255	128	9.07	7.03
511	256	10.3	8.02
1023	512	12.25	9.01
2047	1024	13.4	10.01
4095	2048	14.02	11



Linear
2500
2000
1500
1000
500
0
Search Cost
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Nodes



Random
16
14
12
10
8
6
4
2
0
Search Cost
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Nodes



As we have discussed earlier, perfect Binary Tree's average cost is $O(log_2 n)$, where for a linear tree it is $O(n)$. The graphs here also confirm it.

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: Pratik Patel Date: 03/031/2019