Sameep Shah
COSC 20803
Lab 2
November 1, 2023

I created a method to make a degenerate binary tree. Using the newly created method and the method given already to create a pseudo-random binary tree, I calculated the average number of comparisons and time taken to create both trees with increasing number of nodes(n). The following table (Table 1) contains the relevant data I found in my experiment.

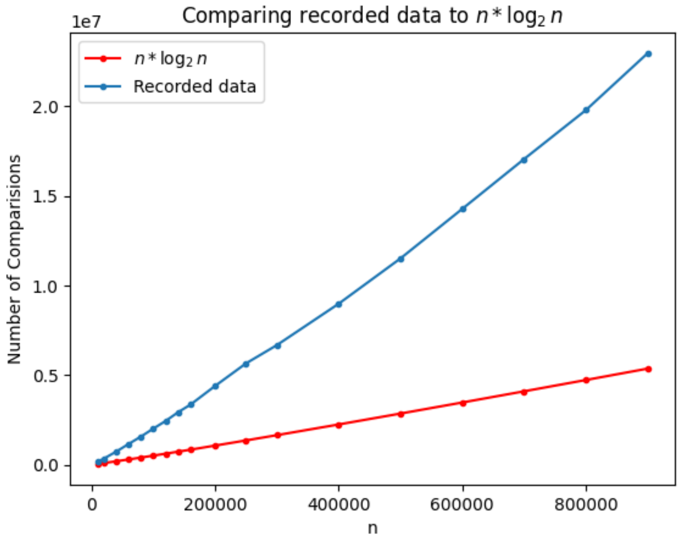
| | **Pseudorandom** | | **Degenerate** | | |
|---|---|---|---|---|---|
| n | Avg. Comparisons | Avg. Time | Avg. Comparisons | Avg. Time | |
| 10000 | 159130.2 | 0.00364288 | 49995000 | 0.10846658 | 314.176693 |
| 20000 | 333638.8 | 0.00635666 | 199990000 | 0.42362914 | 599.4206909 |
| 40000 | 725536 | 0.01261011 | 799980000 | 1.66839257 | 1102.605522 |
| 60000 | 1150150.8 | 0.01959351 | 1799970000 | 3.68926153 | 1564.986087 |
| 80000 | 1565057.6 | 0.02742359 | 3199960000 | 6.65928987 | 2044.627623 |
| 100000 | 2009421 | 0.03568952 | 4999950000 | 10.2274128 | 2488.254079 |
| 120000 | 2436244.2 | 0.03995853 | 7199940000 | 15.1167226 | 2955.344132 |
| 140000 | 2921373.2 | 0.04957737 | 9799930000 | 20.488449 | 3354.562847 |
| 160000 | 3344983.8 | 0.05413482 | 12799920000 | 26.8419461 | 3826.601492 |
| 200000 | 4391807.8 | 0.06689136 | 19999900000 | 42.0170999 | 4553.910579 |
| 250000 | 5651018.8 | 0.07742932 | 31249875000 | 66.9160058 | 5529.954174 |
| 300000 | 6671875.4 | 0.09129846 | 44999850000 | 97.7613509 | 6744.707792 |
| 400000 | 8983097 | 0.11682763 | 79999800000 | 172.10702 | 8905.592359 |
| 500000 | 11511751.5 | 0.17483994 | 1.25E+11 | 256.96638 | 10858.44756 |
| 600000 | 14278595 | 0.24780038 | 1.8E+11 | 383.255602 | 12606.26133 |
| 700000 | 17059353 | 0.26492551 | 2.45E+11 | 504.102046 | 14361.60269 |
| 800000 | 19792894 | 0.28153996 | 3.2E+11 | 642.166966 | 16167.39826 |
| 900000 | 22970496 | 0.36612096 | 4.05E+11 | 852.667513 | 17631.2932 |
| 1000000 | 25609553 | 0.43071888 | 5E+11 | 1060.74193 | 19523.94483 |

Table 1: Experiment data containing average number of comparisons and time required to create a pseudo-random and degenerate tree plotted against different values of n

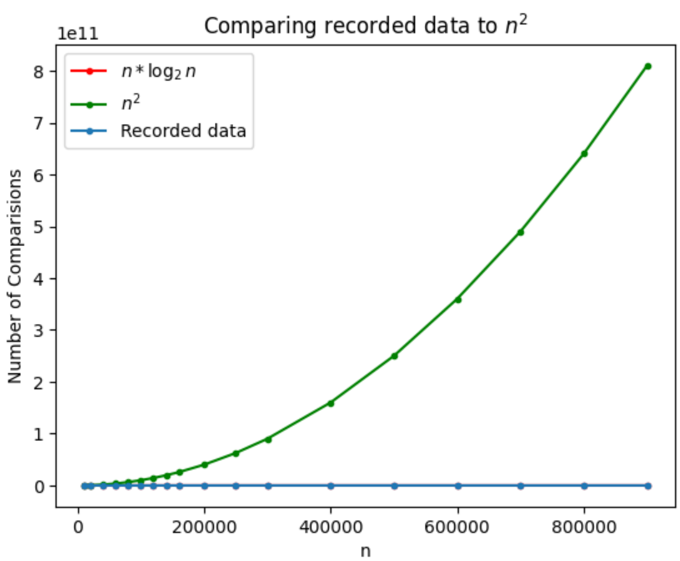
Using the data in Table 1, I plotted some graphs to help us analyze my findings.

Psuedo-random binary search tree:

Graph 1 – Comparing number of comparisons vs n, to a regular function which outputs n * log(n) values.

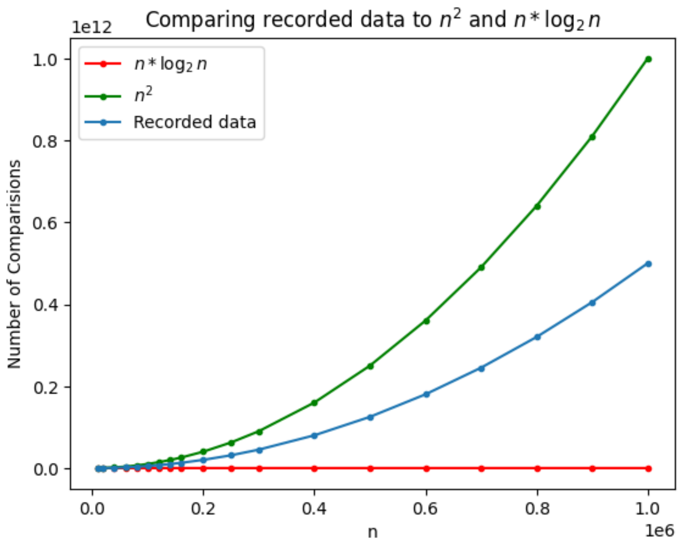



Graph 2 – Comparing number of comparisons vs n, to a regular function which outputs $n^2$ values.

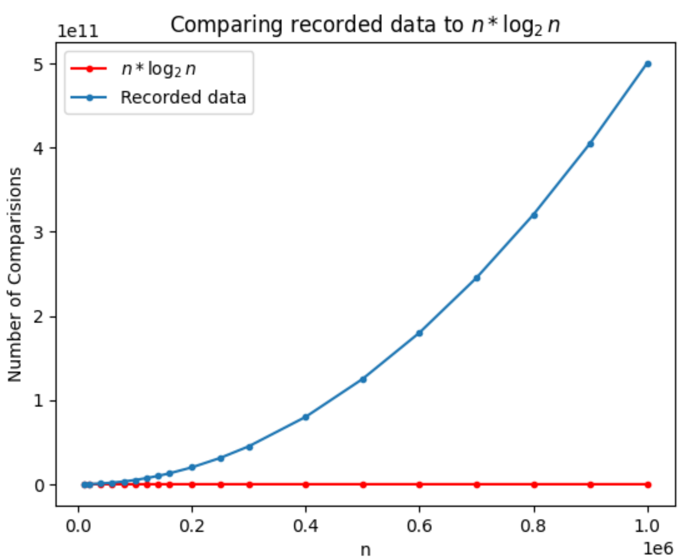


Degenerate Binary Search Tree:

Graph 3 – Comparing number of comparisons my experiment took to functions which output $n^2$ and nlog(n) values.




Graph 4 – Comparing my recorded data to a function which outputs nlog(n) values.



Summary: From graph 1, we can observe that the recorded values for the pseudo-random BST are very similar to a regular function that outputs nlog(n) values. It seems like the values are just a factor more than the regular function. This can also be seen by looking at graph 2, where my recorded data curve and regular nlog(n) curve both dwarf in front of a $n^2$ curve, which can help us realize that the insertion of a new value in a BST should be O(nlogn), as out data tends to nlogn rather than $n^2$.

Similarly, if we look at graph 3, we can observe that my recorded data of inserting a new value into a degenerate BST tends more to a normal $n^2$ function, than a normal nlog(n) function. We can make sure that, this is true if we look at graph 4, where the normal nlog(n) function dwarves in front of my recorded data from Table 1 of inserting a value in a degenerate BST. This makes us realize that in a degenerate BST, inserting a new node will take us $O(n^2)$ time.