Warren Scipio

Lab 6 Report

Organization

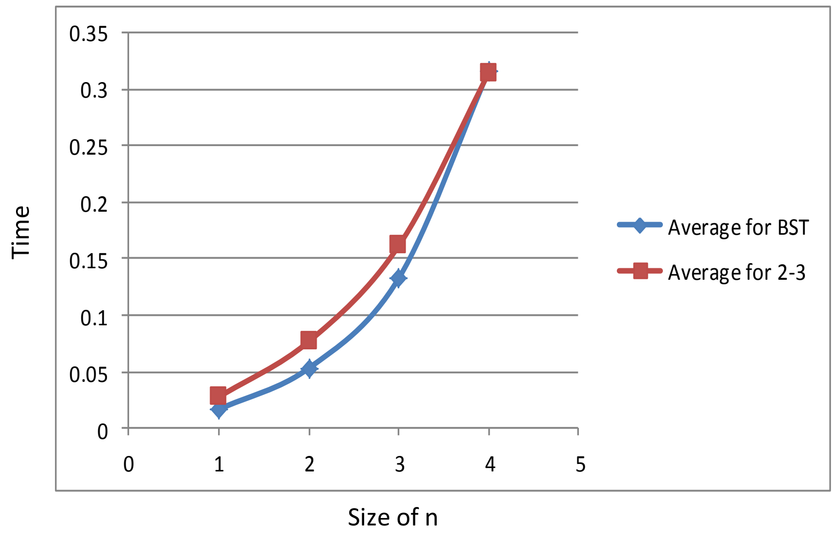
Well I started making the 2-3 tree first to do that I started with a basic main file that would make calls from the file Two3tree. Two3tree was a nightmare to say the least. I tried to start with the insert function and quickly found out I needed more and more little functions to make it work. I built a queue to get level order working properly. In the end I didn't get deletion working so I removed it so it would compile. Sense I only got the insert and not deletion I only tested the time of the insertion of both data structures. I set up the timing portion of them lab by just setting the 4 different n's to an array and cycling through them testing them in one at a time. Changing srand with the next for loop. I collected them and added up the values to get averages for each structure at each n.

Times for insertion

	n=50,000	n=100,000	n=200,000	n=400,000
Average for BST	0.0165334	0.0529772	0.132681	0.316523
Average for 2-3	0.0274348	0.0766028	0.161481	0.314742

Conclusion

Well from my results you can clearly see as n get bigger both data structures take longer to insert all of the data. When comparing the two it seems, according to my data, that the regular binary search tree out performs the 2-3 tree at lower values of n. We only see the 2-3 tree out preform the regular binary search tree when n is 400,000. If the tread stays the same we'd expect to see the 2-3 tree continue beating the binary search tree as n grows even larger. As a conclusion the 2-3 tree is hard to keep track of but I only got to test its insertion, but it did pretty good.



Key for n

1=50,000

2=100,000

3=200,000

4=400,000