

Daniel Darcy
Gupta 1:00 TH
Assignment 4
March 26 2018

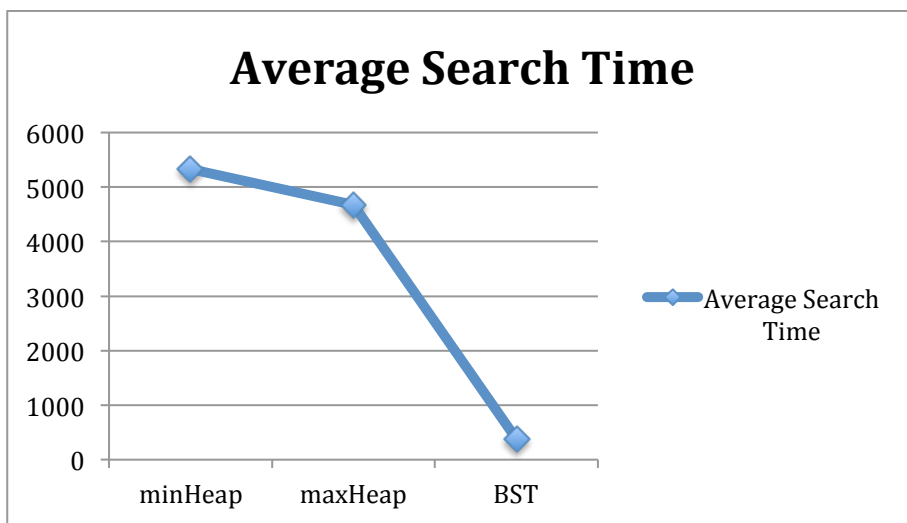
Data Structures and Time Complexities

Over the last few weeks I have been writing a java program. The program focuses on three different data structures; min-heap, max-heap, and binary search tree. The program starts by asking the user for two different file names to be scanned into 2 different arrays. Once these files are scanned in the program then stores the data from the first array in each data structure. It then checks the second array for different values to search for in these data structures. Once these objectives are complete the program prints a detailed report of the different time complexities for each data structure.

When building this program I was under the impression the binary search tree would be the best and most efficient way to arrange all of this data. It turns out that is not correct any of the different data structures are actually pretty efficient. For each data structure the build time is $O(n)$.

After examining the data produced by my program it looks as if the binary search tree was the fastest for searching. It had about $1/10^{\text{th}}$ of the search time of the other two data structures. I believe this to be slightly inaccurate considering the data structures are so similar. Most likely it is because I measured the time complexity slight different for the binary search tree then I did for the min-heap and max-heap because of the way I built the program. Check the plots below for more detailed info on the 3 different data structures.

I believe any of these 3 data structures are efficient but all have their specialties for different methods. I would have to say it is hard to say what data structure is best unless one knows how they will be accessing the data contained in the data structure.



Time Complexities

Average Search Time	
minHeap	5334
maxHeap	4669
BST	374