Tanner Crane

Cpts 233 – Advanced Data Structures and intro to algorithms

Professor Nadra Guizani

A. Problem Statement

The goal of this problem is to take an input file of integers and sort them into a LinkedList data structure. While doing this, we are benchmarking the performance of the algorithm by taking statistics on the speed of each to find the min, max, and med of the list, and the total execution time.

B. Algorithm Design

In my code, I first read in the file using BufferedReader and initially store the data into an ArrayList, after this I sort the array (Using Collections.sort(arr);) and then store them into a LinkedList. I used System.nanoTime() before and after the for loops that add the elements into the data structures, and also during the equations to calculate the med, min, and max to determine the start time and end time of each operation. I calculate total execution time by taking endtime minus starttime. To find the min value I used arr.get(0) and to find the max I used arr.get(arr,size(0) – 1) both these lines of code were stored into min and max variables respectively. Finding the med was a little trickier, so I put two conditions in an if and else block.

C. Experimental Setup

Machine Specifications –

Processor: AMD Ryzen 5 2400G with Radeon Vega Graphics 3,60Ghz

Installed Memory (RAM): 8.00 GB (6.91 GB usable)

System Type: 64-bit Operating System, x64-based processor

Hard Drive Type: Local Disk Drive and Standard Disk Drive

I repeated the experiment 5 times before reporting my final timing statistics in nanoseconds.

O/S and environment used during testing: Windows with the compiler VScode.

D. Experimental Results & Discussion

Attempt #5 of running the code:

Min value: 1

Median value: 2056

Max value: 4000

Total time to insert values into the list: 621 ns

Time to find min: 301 ns

Time to find max: 500 ns

Time to find median: 3800 ns // It makes sense that finding the median would take a little longer than the max and min because the algorithm to find it is more complex.