Justin Lee

CptS 233

Lab report:

Problem Statement:

The main goal of this problem is to test with the benchmarking the performance of a linked list. The results should contain the running time that it takes to read from the text file and find the values that is in the specific location with the finding times.

Algorithm Design:

To solve the problem, I used the linked list from the java library. First, I need to use scanner to accepts the values from the text file. And insert into the linked list. The main consideration of this problem was the adding a value into the sorted linked list. I was guessing using my own linked list will makes more easier to solve, but I want to try with the linked list from the java library. Anyway, I used the binary search finds the right place to insert the value into the linked list. To use binary search, I needed to create the simple compareTo method to compare the two integer values. By the using binary search algorithm, I was possible to insert the values as the sorted. After the finishing those tasks, it needs to compute the out values of maximum, minimum, median, with the program running time to computes them. Because it is a sorted linked list, the minimum and maximum would be located at the 1st and last of the lined list. And the median located at size / 2 if the size is odd or average of two middle values if the size is even. And also compute the running time for insertion. At last, I used the nanotime method to computes the run times. The run time should write as microseconds, which was nanoseconds / 1000.

Experimental Setup:

The specification of machine that used in this assignment was Window 10. HP Pavilion x360 Convertible 14m-ba0xx. Intel® Core™ i5-7200U @2.50GHz, 2712Mhz, 2Core(s), 4 Logical Processor(s). Ram available is 2.13GB. Hard drive type is solid state drive and the model is SanDisk SD8SN8U-128GB-1006. Used compiler of Eclipse IDE for the java Developers. I tested the program more than 5 times, and the results was different each time.

Experimental Result & Discussion

There were two text files given to test the program. The difference between two text files was the size of the data. The second text file contains much bigger size of data than first text file. I tested the program 5 times for each text file and here is the result. For the first text file, the average time for max is 4.6 microseconds, the min is 1.6 microseconds, med is 67.4 microseconds, and insertion time is 36002.4 microseconds. And the second file for max is 5.4 microseconds, min is 1.2 microseconds, med is 2359 microseconds and insertion time is 111342821.2 microseconds. There was not much difference from the max and min because the method running algorithm didn’t change by the size of data but computes the running time for med and insertion change by the size of data. It represents min/max algorithms are O(n) execution time and insertion/med algorithms are doesn’t.