Implement a **MERGE** sort and **BUCKET** sort as defined in class. Input file consists of integers, one integer per line. Input file name is to be read from command line. For bucket sort, the range of the input numbers, can be obtained after reading the contents of the file. The minimum and maximum number can be determined after reading the contents of the file.

Output should be printed upon request in the command line, based on user entered range of sorted numbers. The range of numbers to be output on the command line is requested to the user as shown below. **Only** the sorted numbers in that range (inclusive) are to be output on the console. Two different programs, one for merge sort and one for bucket sort should be submitted along with Readme.txt if necessary.

**Input example:**

34

567

12344

122

3

45

**Sample Program Execution:**

Enter file name: input.txt  
Enter left range: 2  
Enter right range: 4

Sorted numbers in the selected range are:

34  
45  
122

**Plots:**

Modify the above bucket and merge sort programs to calculate the elapsed time (in nanoseconds) for different input sizes. The input size should start from 1000 numbers and ends at 100,000 incrementing the input size by 1000 every time. The inputs given here are random numbers.

Plot all the execution times of both Merge sort and bucket sort for all the input sizes from 1000 to 100000 in an excel spreadsheet. In the same plot, also add the asymptotic time for merge sort and bucket sort.

**Plot in excel:**

**X axis:** input size: 1000, 2000, 3000,……………….100000;

**Y axis**: execution time in nano seconds

**What to submit:**

A source code for each algorithm (merge sort and bucket sort) and Readme (if necessary) using “handin”. Upload plot file into “Assignments” on Canvas.