CS 325 HW 1 – 30 pts

Submit a copy of all your code files, data.txt and the script file in a ZIP file to TEACH. Submit the written answers to Problems 1, 2 & 5 in Canvas.

Problem 1: *(5 pts)* The Mergesort3 algorithm is a variation of Merge sort that instead of splitting the list into two halves, splits the list into three thirds. Mergesort3 recursively sorts each third and then merges the thirds together into a sorted list by calling a function named Merge3.

a) Write pseudo-code for Mergesort3 and Merge3
b) Let T(n) denote the running time of Mergesort3 on an array of size n. Write a recurrence relation for T(n).
c) Solve the recurrence relation to obtain the asymptotic running time.

Problem 2: *(5 pts)* Stooge sort is a recursive algorithm for sorting a list of numbers. The algorithm is described as follows:

- If the value at the start of the list is larger than the value at the end, swap them.
- If there are 3 or more elements in the list, then recursively,
  - Stooge sort the initial 2/3 of the list
  - Stooge sort the last 2/3 of the list
  - Stooge sort the initial 2/3 of the list again

In the recursive calls round the 2/3 *upwards*, e.g. rounding 2/3 of 5 should give 4 rather than 3. Otherwise the list may not be properly sorted for some list sizes.

a) Write pseudocode for Stooge sort.
b) Let T(n) denote the running time of Stooge sort on an array of size n. Write a recurrence relation for T(n).
c) Solve the recurrence to determine the asymptotic running time.

Problem 3: Implementation of Stoogesort and Mergesort3 *(10 pts)*

Implement mergesort3 and Stooge sort to sort an array/vector of integers. You may implement the algorithms in C, C++ or Python, name the programs "mergesort3" and "stoogesort". Your programs should be able to read inputs from a file called "data.txt" where the first value of each line is the number of integers that need to be sorted, followed by the integers.
Example values for data.txt:

4 19 2 5 11
8 1 2 3 4 5 6 1 2

The output will be displayed to the terminal.
For the above example the output would be:

2 5 11 19
1 1 2 2 3 4 5 6

Test your code on the school's server, flip. Your code should run with the HW1.sh script.

CS 325 HW 1 – 30 pts

Problem 4: **Running times** *(5 points)*

Modify code- Now that you have verified that your code runs correctly using the data.txt input file, you can modify the code to collect running time data. Instead of reading arrays from the file data.txt and sorting, you will now generate arrays of size n containing random integer values from 0 to 10,000 to sort. Use the system clock to record the running times of each algorithm for ten different values of n for example: n = 5000, 10000, 15000, 20,000, …, 50,000. You may need to modify the values of n if an algorithm runs too fast or too slow to collect the running time data. All running times should be greater than 0 and less than 20 seconds. Output to the terminal the array size n and time to sort. Name these new programs merge3Time and stoogeTime.

Submit the code to TEACH.

Problem 5: **Data analysis** *(5 points)*

a) Collect running times - Collect the timing data from running your programs on the engineering flip server. You will need at least ten values of t (time) greater than 0. If there is variability in the times between runs of the same algorithm you may want to take the average time of several runs for each value of n. Create a table of running times for each algorithm. Is this best case, worst case or average case running time? Did you take the average of multiple runs?

b) Plot data and fit a curve - For each algorithm plot the running time data you collected in part a) on an individual graph with n on the x-axis and time on the y-axis. You may use Desmos, Matlab, R or any other software. What type of curve best fits each data set? Give the equation of the curve that best "fits" the data and draw that curve on the graphs. How does your experimental running times compare to the theoretical running times?

Note: All code should be fully commented and include your name, date and class. If you use any resources when writing the code, cite them in the comments.