COSC581 - Algorithms
Spring 2023
Homework #3

Due: Tuesday, 02/14/2023, before class.

1. Use the master theorem to solve the following recurrences. Show your work for each problem:
    a. $T(n) = T(n/3) + 10$
    b. $T(n) = 4T(n/4) + n/(\log(n))$
    c. $T(n) = 9T(n/3) + n^2\log^3(n)$
    d. $T(n) = 3T(n/6) + n^3$
    e. $T(n) = T(n) + \log(n)$
    f. $T(n) = 2T(n/2) + \Theta(n^2)$

2. Sort the set of integers {489, 783, 66, 1, 42, 88, 1092, 47, 68, 999} in monotonically non-decreasing order using radix sort. Show your results after each pass of the file.

3. Are the integers in this array {4, 6, 5, 7, 3, 11, 8, 9} stored as a min-heap? If not, extract the root and re-heapify until it is. Show each step.

Programming Assignment #1

Due: Tuesday, 02/28/2023, at midnight.

Please note that this programming assignment will be part of the overall homework category, but will be worth the same as two homework assignments. Please email your code and accompanying files to ababjac@vols.utk.edu.

In class we discussed the median-of-medians algorithm, which solves the selection problem in linear time. Implement a quicksort algorithm that uses this method to select the pivot value. Specifically:

- Set r at one of 5, 7, 9, or 11. Justify the decision.
- Implement a straight insertion sort to sort *n* when it becomes small
- Experimentally determine the value of *n* at which quicksort should revert to insertion sort and unwind the recursion. Describe how you made this determination, and the factors upon which it depended.

You may any programming language you like, as long as your code compiles and runs on the EECS lab machines using only software currently installed. A user should see something similar to the following when running your program.

```
UNIX>> ./sort input_file
2 5 8 11 23 34 52
UNIX>>
```

Where input_file is the name of a file containing integers separated by spaces. Your program should read in the elements from the file and print the sorted list to the console. The input will be capped at one million elements.

You should submit a .tar or .zip file containing the following:

- Your code (.py, .cpp, .jar, etc. - no notebook style scripts will be accepted)
- A readme of how to run your code.
- A 1-page write-up explaining your experimentally determined choices of r and n as described above. Graphs and/or tables to support your choice should be included. Please also include any other important implementation specific details about your code.
- A makefile (if your code requires compilation)