**ECE 1111: Engineering Computation I**

**Homework No. 11: Lists and Trees in C and Python**

**Deposit your work in:**

**/data/courses/ece\_1111/current/homework/hw\_11/<lastname\_firstname>**

**Goal:** The design of a data structure can be crucial to the efficiency with which you can manipulate data. In this assignment, you will compare the efficiency of two sort algorithm implementations.

**Description:** Create a text file with random text data that is 10,000,000 lines long (use wc to verify this). For example, use the commands:

**cat /usr/include/\*.h > temp.txt**

**cat temp.txt temp.txt >> temp\_2.txt**

**mv temp\_2.txt temp.txt**

**[... repeat ...]**

until you have a file that is long enough.

1. (/p01) Sort this file using the Unix sort command and time the execution time using the Unix time function (Google search it).
2. (/p02) Write a Python program that reads this file into memory and sorts it – about two lines of Python code ☺. You can use the built-in capabilities for list management and sorting in Python.
3. (/p03) Write a C program that sorts this data, line by line, in lexical order using the Unix binary tree utility (‘man tsearch’) and a linked list. You can use whatever linked list implementation you like.

Time the performance of all three versions using the Unix time function. Comment on the pros and cons of each approach. Your analysis need not be long – a few paragraphs should suffice. Your code must compile and link using make. You will be graded on both the structure and functionality of your code as well as the quality of the documentation. In a file AAREADME.txt, provide an analysis of the run-time efficiency of the three approaches. Keep your explanations simple and to the point.