PROJECT#2

[SORTING]

Due Date: OCT 15, 2006 (800 points)

CS 4/56101 DESIGN & ANALYSIS OF ALGORITHM

Fall 2006, Department of Computer Science, Kent State University

1. In this project you will compare the performances of few sorting algorithms. The sets are (1) **Merge Sort** vs. **Heap Sort** for Contiguous List (2) **Quick Sort** vs. **Radix Sort** for Contiguous List. You need to compare the algorithms in only one of these sets. Pick the set based on your random CID. If your CID is odd pick set 1 otherwise pick set 2.

   (a) Implementation (300 points): Given a text document you want to sort the words according to their dictionary order. Each of your implementation should be able to read first **n** words from a text file containing 1000-10,000 words. (get a sample "random.txt" file from the Web-book). You can assume all words to be of less than 15 alphabets. Your implementations should be able to read, sort and finally print back the sorted result in another file. If specified it should be able to iterate the sorting operation **r** number of times. Your algorithm should keep account of the sorting time and at the end display summary statistics showing (a) time spent in assignment (b) time spent in comparison (c) time spent in I/O (d) total time spent, (e) total count of assignments (f) total count of word comparisons. (g) number of words sorted. A typical command line invocation should look like this:

   %mysort myinput.txt myoutput.txt n r

   (b) Report (300 points): Generate your own data files and prepare the following three graphs comparing the algorithms in your set. The three graphs should respectively show (i) number of comparisons (ii) number of assignments and (iii) total execution time (y-axis) for various **n** (x-axis). Explain your answer by comparing the plots with respect to the performance analysis derived in the class.

   (c) Take the data files "sorted.txt", "reversesorted.txt", "partsorted.txt". Plot and compare the running time performances of your algorithms for each of these three files. (200 points).

*If you have question about the sorting order, see sorted.txt. It has been created using the MS Word's table sort function.

For submission follow the same guideline as Project#1.