CSCI 665 Foundations of Algorithms

Mohan Kumar

SPRING 2014

Lab Assignment I

Assignment Problem

MY_CHOICE_QSORT: The Quicksort algorithm is an efficient and popular sorting technique that sorts a list of keys S[1],S[2], . . ., S[n], recursively by choosing a pivot key. The best-case running time of Quicksort [3] is $O(n \log_2 n)$ and its worst-case running time is $O(n^2)$. Several improvements and modifications have been proposed to improve Quicksort's worst-case behavior. For example, the paper by Wainwright [1] presents Bsort, a variation of Quicksort that combines Bubble-sorting techniques with the Quicksort algorithm. You will find other methods in [2,3,4] as well as a Randomized algorithm in Chapter 13 of the textbook. Alternatively, you can consider a method not listed above, but available in the literature; please include the reference in your report. Please choose ONE improvisation of Quicksort (of your choice) – let's call it MY_CHOICE_QSORT.

Write programs to implement sorting algorithms using the following methods:

1. MY_CHOICE_QSORT
2. Quicksort
3. Mergesort
4. Heapsort

Execute your sorting programs for the following sets of data, all real numbers:

a. Set_1: Ordered List
b. Set_2: Random List
c. Set_3: 25% of the List sorted
d. Set_4: Poisson distribution of data values; The probability function [5] is given by $f(k;\lambda) = Pr(k) = \frac{\lambda^k e^{-\lambda}}{k!}$; e = base of the natural logarithm = 2.71828…; use $\lambda = n/2$, where n is the number of data elements; k varies from 1 to the number of data elements in the set (1000, 10K, 50K, 500K and 1M).

Presentation of Results: Measure CPU time, number of partitions (only for Quicksort and, MY_CHOICE_QSORT) and number of comparisons for data sizes 1000, 10K, 50K, 500K and 1M. Present your results using tables or graphs and write a 1-page report. The plots (graphs) should be accompanied by brief explanations about the characteristics of the algorithm and the data sets. The report should have a psuedocode for MY_CHOICE_QSORT and summarize the behavior of all Sorting algorithms tested.

Required Submissions: a one page summary report including the algorithm, the code, sample data and code to generate data, results* in tables, and instructions for executing your code.

Presentation of Results*: Measure CPU times/memory usage, compare results and use tables to present results.
*Results: Each data point should be an average of 10 or more reruns of the execution.

Programming Language and Data Structure: **Your choice**. Please mention your choices and provide a justification in the summary report.

Assignment submissions are due by 9AM March 06, 2014. Submission instructions will be posted on myCourses.

PLEASE NOTE THAT your PROGRAMS WILL BE TESTED WITH DIFFERENT DATA SETS.

References

[1] R.L. Wainwright, A Class of Sorting Algorithms based on Quicksort, Communications of the ACM, Vol. 28, No. 4, April 1985, pgs. 396-402.
[2] C.R. Cook, and Kim D.J, Best sorting algorithm for nearly sorted lists, Communications of the ACM, Vol. 23, No. 11, Nov. 1980, pgs. 620-624.
[3] C.A.R. Hoare, Algorithm 64: Quicksort, Communications of the ACM,Vol. 4, No. 7, July 1961, pg. 321.
[4] [3] M.N. vanEmden, Algorithm 402: Increasing the efficiency of Quicksort, Communications of the ACM, Vol. 13, No. 11, Nov. 1970, pgs. 693-694.
[5] http://en.wikipedia.org/wiki/Poisson_distribution