CSCI 665 Foundations of Algorithms.
Lab assignment I
Spring 2019

For the Lab Assignment, you are required to submit on MyCourses. Dropbox has been enabled for submissions.

- Summary report (Maximum 1-page)
- Your algorithm
- Your code
- Sample data and code to generate data
- Results* and
- Instructions to execute your code.

Include all the above in a zip file and submit in the Dropbox on or before **4:30 PM on 02/15/19**.

*Presentation of results: measure CPU time taken for increasing data sizes. Use tables or plots to present results for increasing data sizes. Consider real numbers for input data. Each data point in your tables or plots should be an average of 19 or more reruns of the code.
Penalty for late submissions: **25% PER DAY**; For example, if you submit after 4:30 pm on 2/15/2019, but before 4:30 pm on 2/16/2019, your project will be graded for 75 % of max points.

Problem description

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.

Implement MY_CHOICE_QSORT algorithm and the quicksort algorithm discussed in class. Execute your sorting programs for two types of data sets (real numbers). Results should include comparisons of the two algorithms for the suggested data sizes.

a. Set_1: Random Lists (1000, 10K, 50K, 100K and 500K)
b. Set_2: 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 n, the number of data elements in the set (1000, 10K, 50K, 100K and 500K).

[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] M.N. van Emden, 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.