Project Propsal

Authors:
Joshua Hernandez [umhern23@myumanitoba.ca]

Paymahn Moghadasian [umpaymah@myumanitoba.ca]

Instructor: Dr. S. Durocher

COMP 4420
Advanced Design and Analysis of Algorithms

Due : Febuary 24, 2014

Project Proposal

Sorting is a well studied topic in algorithms, in particular quicksort has been extensivly studied. ***Continue the intro****

A lot has gone into the creating faster sorting algorithms.

During the ICALP 2013 conference there is a paper that explors dual partition quicksort. They study the various parititon algorithms that can be applied and their assosiated asymptotic runtime. They focus on array swaps and element comparisions to determine the asymptotic runtime. They are able to mathemetically generalize the notition of dual parititon quicksort regardless of the partition algorithm used. From this they are able to determin that the optimal dual quicksort will run in $1.8n \log n + o(n \log n)$ time. Next the experement with various quicksort implementations, each using a different partitioning algorithm.

We hope to verify what the previous paper has established. We then also wish to expand the scope of the original paper. We will implement quicksort using various partitioning algorithms, again using varying number of partitions. We hope to determine if varying the number of paritions will yeild to a more efficent quicksort algorithm.

In parallel we also hope to mathematically describe multi-partition algorithms, based off the proof given in the paper. During this process, we hope bring to light all the problems that may arise from this generalization.

The goal of this project is to investigate multi-paritioned quicksort and hopefully determine what may the optimal number of parititions could be and bring to light any of the challenges that arise from this analysis. Pariticularly if more than two parititons provide any benifit either in parctice or in theory.

- Week 1
  - Implement basic quicksort
    - Pivot Picking
      - first element as pivot
      - median of first, middle, and last
      - Find others
    - Partitioning algorithm
      - The basic one
    - Each version has an insertion sort flag on arrays with small size
    - Test functionaliy on small arrays
  - Implement 2 parition quicksort
    - Piviot picking
      - first and last elements
      - two middle elements of entires from 5 entries

            - Test with 4 entries as well
        - Paritioning Algorithm
            - Basic Paritioning
            - make smalls then bigs
            - flag paritioning algorithm
        - Each version has an insertion sort flag on arrays with small size
        - Test functionality on small arrays

- Week 2
    - Implement 3 and 4 parition quicksorts
        - Parition algorithm
        - Basic Parition
        - Look for parition algorithms
        - Each version has an insertion sort flag on arrays with small size
        - Test functionallity on small arrays

- Week 3
    - Run several experinments using all the quicksort algoirthms implemented
    - Run several arrays sizes
    - Array 'types'
        - sorted arrays
        - reverse sorted arrays
        - random arrays
        - partially random arrays
    - Preliminary analysis of data

- Week 4
    - Analyze data
    - Write paper
    - Make Presentation