Eric Furukawa

CPTS 223

Section 01

Programming Assignment 2 Report

A: Problem Statement

In this exercise, different algorithms designed to complete the same task were tested on varying sizes of data to show the efficiency of them and the growth of these efficiencies over growing data sizes. Different algorithms may grow linearly, exponentially, or otherwise as we have discussed on class according to their algorithmic design.

B: Experimental Setup

PC Specs:

Type: Laptop

**OS: Windows 10 Home**

Processor: Intel (R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHZ

**RAM: 12.0 GB (11.9 usable) (12288 MB)**

**Clock speed: (2.7GHz)**

**CPU: 4 CPUs**

System Type: 64 bit OS, X64 based processor

Condition: About 3 Yrs old (heavy use)

**Compiled and tested using PuTTY(linux) g++**

The data sizes 8, 16, 32, 64, 128, 256, 512, 1024, 20481 4096 and 8192 were each tested with 10 unique data samples for time recording. These were averaged and plotted for each equation in the graphs.

C: Experimental Results

Plots (Two extra provided to see growth of other algorithms):

Observations:

The results of the PA’s experimentation and trials matched my theoretical expectations of the different algorithms. Through class we have discussed how loops in algorithms and iterations impact performance over amounts of data to be taken in. Only one of the sums had linear growth. The some of the algorithms may appear to be linear in the first graph, but this is due only to the sheer magnitude of MaxSubSum1. In general, the length of execution increases for the more for loops in the code. Un-nested for loops will increase the time more linearly while nested loops will start to increase the function’s growth in a much sharper rate the more the data size is increased. Algorithm 1 must iterate through 3 nested loops while algorithm 2 must iterate 2 nested loops. Algorithm 3 recursivly iterates through 2 non-nested loops and algorithm 4 has only one for loop. My theoretical estimations were that alg1 would be about n^3 efficient, alg 2 would represent a quadratic n^2 efficiency, alg3 would represent somewhat in between alg 2 and 4 for efficiency as the loops aren’t nested but are recursively iterated through, and alg 4 represents a linear growth. Our data reflected this in the comparison of the differing algorithms. Alg1>Alg2>Alg3>Alg4, which all but alg 4 represented quadratic curves (and alg 4 representing a more linear growth).