

# **PA3: Bucket Sort (Sequential)**

**Evan Grill**

**CS 415**

**March 30, 2017**

## Problem

This project asked us to implement a bucket sort sequentially.

## Procedure

This project uses generated input files formatted with a value on the first line, which represents the size of the data set, followed by successive lines containing that many ints. The program first reads in the file into a single dimension vector. It then sequentially moves through the elements of the vector copying them into appropriate buckets. For this project I used 10 buckets, but that is easily varied by changing a defined macro in main.cpp. The buckets are divided up into equal ranges by dividing the largest number by the number of buckets. After the buckets are filled, `std::sort` is run on each bucket individually. After sorting, each bucket is copied back to a single dimension vector.

## Data

Data Size	Execution Time
10	0.0000119209
100	0.0000555515
1000	0.0002472400
10000	0.0013322800
100000	0.0074834800
1000000	0.0795467000
10000000	0.8607750000
100000000	9.2976700000
1000000000	102.1200000000
1250000000	131.6520000000
1500000000	176.1950000000
1750000000	255.3590000000
2000000000	331.7810000000

Table 1: Execution time (in seconds) of sequential bucket sort.

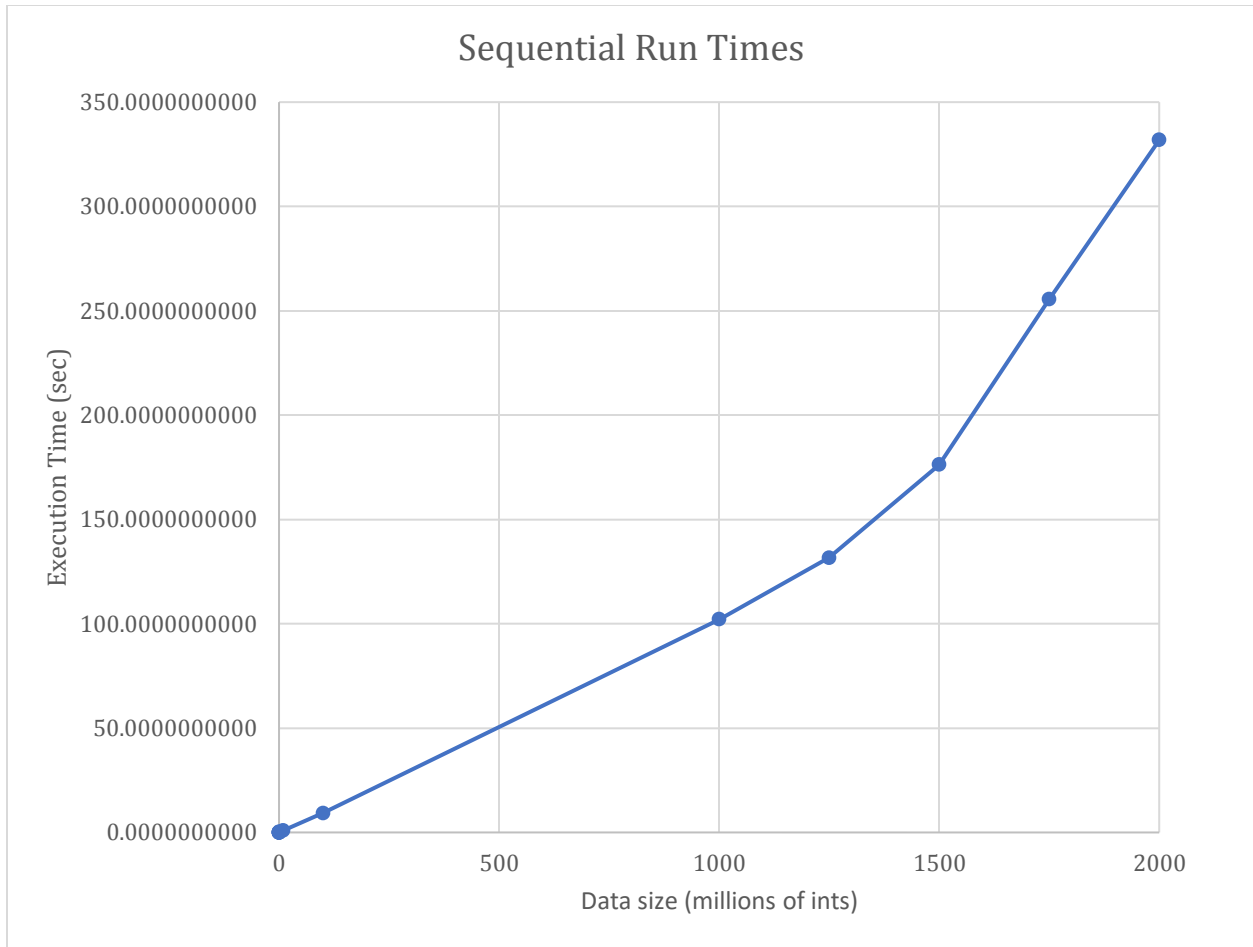


Figure 1: Chart representing the data from Table 1

## Results

The project presently only has the sequential method implemented, so there is not any data to compare it to, but it displays an expected linear growth rate until 1.25 billion ints. After this point, it starts to curve upwards, in what I suspect is evidence of thrashing.

The next portion of the project and report will add parallel implementation with comparisons to sequential, speed up factors, and efficiencies, for multiple amounts of parallel processors.