Main sorting algorithms: QuickSort

Used a loop for the for the quicksort instead of recursion to compare the time complexities of the 2 and the amount of data that can be processed with each algorithm.

Introduction

• Background

• Project Rationale/Theme/Objectives

The program takes in data from gas wells and processes the daily flowrates of the well to be used to track production as well as possible degradation of the well. Comparing the well flow rate to temperature and pressure you could also signs the well freezing off or if pressure is subsiding and temp is increasing the well could be flowing an increasing amount of water in place of gas.

• What data structures / algorithms did you use?

Quicksort adapted for multidimensional data

multidimensional arrays

• Measurement / Algorithm Design

o What did you measure?

Measured the cpu usage and the memory usage as there wouldn’t be a significant change in the O(nlogn) of the existing quicksort algorithm but a change in the number of operation and stack usage depending on the size of the data being used.

o How did you measure?

Used the performance profiler in Visual studio to see CPU and Memory usage of the program to compare to other algorithms.

o Design choice made for each algorithm in terms of time/space complexity

Main function reads into map, transfers to vector<GasWEll>

Sorts vector <GasWell> based on quicksortFlow

comparebyFlow

O(nlogn) for average complexity on quicksort. Worst-case of O(n^2)

• Empirical Analysis

o How do you know what you did was correct?

By finding how long the program ran for the individual quicksort function element and comparing that to the theoretical time complexity number calculated.

o Generate test cases for your data structures and algorithms, varying input sizes

tested with 80, 100, 1000, and 10,000 pieces of data.

o You may present your findings with visualizations, graphs, and analysis of performance trends

• Findings/ What did your results reveal?

• Conclusion / limitations

o What surprised you (if anything)?

o What did you learn?

• Include code repository details (if applicable)

• List of individual contributions / Workload distribution