|  |
| --- |
| Carlow IT |
| Quick Sort Report |
| Comparison between Serial and Parallel sorting |

|  |
| --- |
| (Student) - Ignas Rocas, C00135830  12-18-2021 |

# Abstract

The document describes test that have been conducted using code provided by Lecture, Joseph Kehoe. The program implements quick sort algorithm in c++ programming language.

The test consists of running the quick sort algorithm using “pragma omp” which manages the threading automatically for the programmer with random values generated from 0 to 50’000.

# Test

Table 1 display results of the test as described previously. “Number of samples” represent the number random values generated in the array; “serial time” describe time it took to complete run the test without “pragma omp” whereas “parallel time” shows the time it took for the test to be completed in seconds with the use of “pragma omp”.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number of samples: | 10’000’000 | 1’000’000 | 100’000 | 50’000 | 20’000 | 10’000 |
| Serial time in seconds: | 13.9345 | 0.3286 | 0.0241 | 0.0121 | 0.0035 | 0.0016 |
| Parallel time in seconds: | 13.2616 | 0.3461 | 0.0237 | 0.0114 | 0.0046 | 0.0024 |

Table 1 - Quick sort time test

# Result

It is clearly show that running the program serially generally is faster when the number of samples are below the 20’00 mark but some anomalies may occur since the algorithm is not static (example of 1’000’000 serial is quicker then parallel). The numbers generated are not the same each time and it all depends on the number of times the algorithm needs to split the array with the pivot.