**Name:** SANDILE NGWENYA

**Topic:** Parallelization of Quick sort or Merge Sort Algorithm

**Language:** C

**Algorithm Definitions:**

1. Quick Sort: it is a sorting algorithm that brings in the concept of Divide-and-Conquer by splitting a large array of data into smaller subsets. This would then mean that with each iteration there’s 3 sequential stages, that is:

* Splitting the input into two components.
* Sorting them accordingly (specific rule mentioned).
* Recombining them.

1. Merge Sort: a sorting algorithm that repeatedly divides the array into half until the subset size is equal to 1. The functions to help merge the subsets are then called/used combining the sorted arrays until whole array is used.

**Sequential aspect:** Both of these functions have a serial concept in them where one stage leads to the next, or in some way a later stage **depends on the previous stage in sorting** the array

**Parallelization vs Sequential**: . In the project sequential codes for both algorithms will be shown written in C. Similarly these serial codes will be parallelized in C then the run times will be compared to see which is the best solution for the algorithms.

**Points to note:** As this would be experimental before giving conclusive judgements there are some points to note, that might influence the output (to be explained in greater detail in presentation):

* Hardware or Software
* Number of programs running
* Scalability
* Amdahl Law how even the lightest serial code has an influence to the runtime