Assignment 5 (Parallel Sorting)  
Program Structures and Algorithms

Spring 2023(SEC –03)

NAME:ALTAF SALIM SHAIKH

NUID:002774748

**Task:**

The task at hand entails the implementation of a parallel sorting algorithm that ensures the sorting of each partition in parallel. Two distinct strategies shall be evaluated for determining the suitability of parallel sorting. The first scheme involves establishing a cutoff value, which shall be updated based on the initial argument specified during program execution. The responsibility of identifying an appropriate value for this cutoff lies with the implementer, who should conduct sufficient experimentation to determine its ideal value. If the number of elements to be sorted falls below the determined cutoff value, the system sort shall be employed. The second approach involves computing the recursion depth or the available number of threads. Based on this computation, the ideal number of separate threads (which must be a power of 2) shall be determined. Thereafter, the number of partitions that should be parallelized will be arranged, with recursion ceasing after the depth of lg t is attained.

**Reports.**Console Output  
Graphical user interface, text, application

Description automatically generated  
  
  
  
***1.Array Size -100,000 cutoff-5000***Chart, line chart

Description automatically generated

***2.Array Size -200,000 cutoff-6000***  
Chart, line chart

Description automatically generated  
  
***3.Array Size -300,000 cutoff-7000***  
Chart

Description automatically generated  
  
  
  
  
 **Relationship Conclusion:**

The performance of the parallel sorting algorithm is significantly influenced by the cutoff value and the number of available threads.

Using more threads generally leads to better performance, up to a certain point.

The optimal number of threads appears to be between 8 and 16 threads for most cutoff values.

The best cutoff value for achieving the best performance varies depending on the number of available threads. However, based on the presented data, it appears that a cutoff value of around 12,000 leads to the best overall performance for most thread configurations.

It is important to experiment and find the ideal combination of these factors to achieve optimal performance.

Overall, the three datasets consistently suggest that the parallel sorting algorithm's performance can be optimized through careful consideration of the cutoff value and the number of threads used.