Program Structures & Algorithms  
Spring 2022  
Assignment No. 4

Name: Urvang Patel  
(NUID): 001568243

•**Task :**

Your task is to implement a parallel sorting algorithm such that each partition of the array is sorted in parallel. You will consider two different schemes for deciding whether to sort in parallel.

1. A cutoff (defaults to, say, 1000) which you will update according to the first argument in the command line when running. It's your job to experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead.
2. Recursion depth or the number of available threads. Using this determination, you might decide on an ideal number (*t*) of separate threads (stick to powers of 2) and arrange for that number of partitions to be parallelized (by preventing recursion after the depth of *lg t* is reached).
3. An appropriate combination of these.

•**Output screenshot:**

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

**•Evidence / Graph:**

**Array Size 20000:**

Graphical user interface, chart, line chart

Description automatically generated

**Array Size 25000**

Chart, line chart

Description automatically generated

**Array Size 29000**

Graphical user interface, chart, application, line chart

Description automatically generated

**•Relationship Conclusion:**

After running the experiments, we can say that

* Ideal number of thread count is 4 because we can see that maximum time gain is in increasing threads from 2 to 4.
* Beyond 4 increasing the threads does not make any significant gains in the run time.
* From the graphs we can see that when cutoff is around one third of the array size, we get the fastest sorting times, with lower number of threads.