Peizhen Wu (NUID : 001567656)

**Program Structures & Algorithms**

**Spring 2022**

**Assignment No. 4(Parallel Sort)**

**Task :**

* 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.
* 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.
* 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).

Implementation of threads:

文本

描述已自动生成

Modify Cutoff==500 and ThreadCount == 2, 4, 8, Array.length==1 million

图表, 折线图

描述已自动生成

Modify Cutoff==1000 and ThreadCount == 2, 4, 8, Array.length==1 million

图表, 折线图

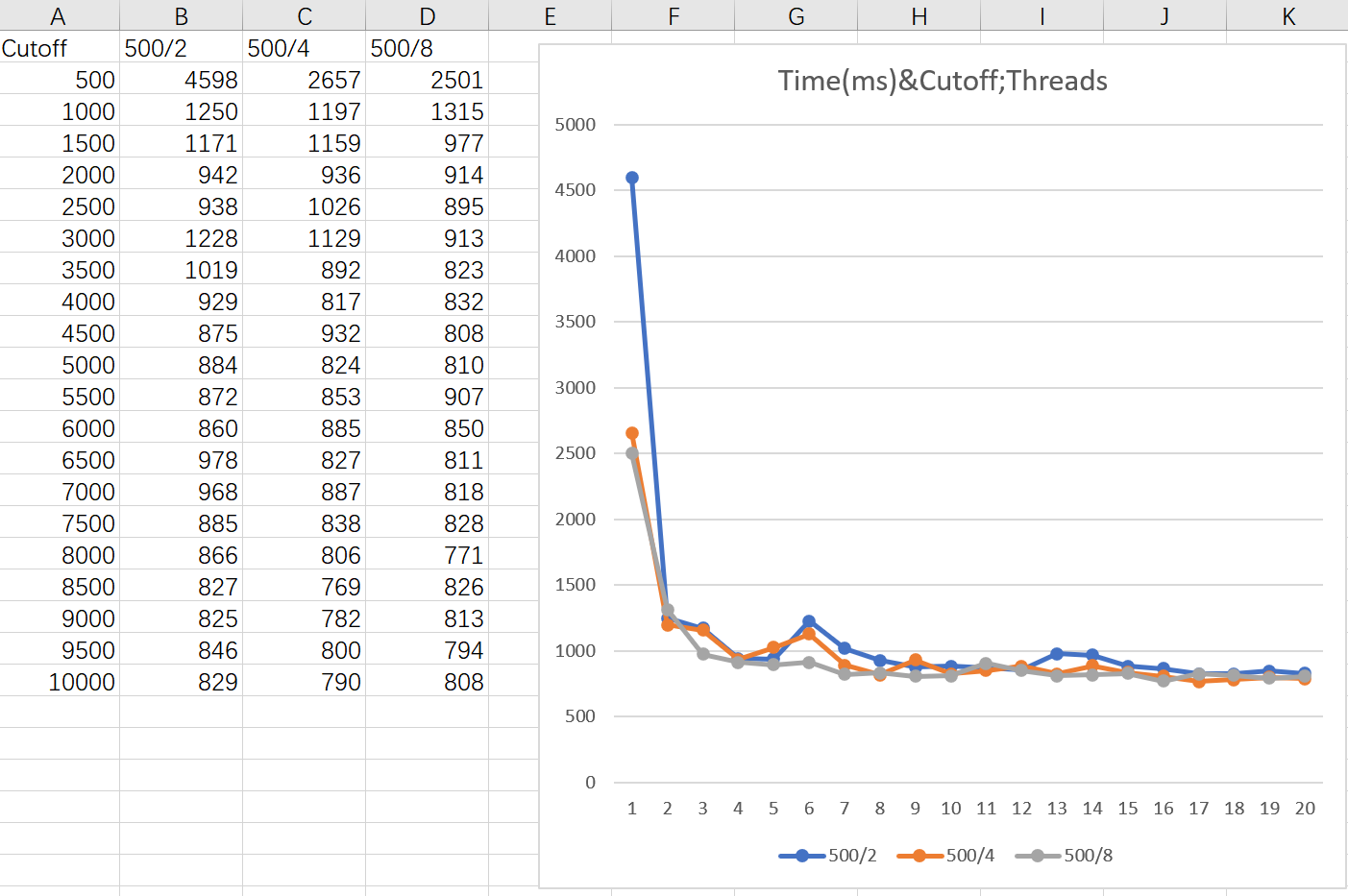
描述已自动生成

Modify Cutoff==2000 and ThreadCount == 2, 4, 8, Array.length==1 million

图表, 应用程序, 折线图

描述已自动生成

Modify Cutoff==500 and ThreadCount == 2, 4, 8, Array.length==**2** million



Modify Cutoff==1000 and ThreadCount == 2, 4, 8, Array.length==**2** million

图表, 折线图

描述已自动生成

Compare 1000 Cutoff&500 Cutoff; ThreadCount == 2, 4,, Array.length==**1** million

图表, 折线图

描述已自动生成

**Conclusion**: From the scatter plots, we were able to come to conclusion that the increasing threadcounts do reduce the cost of time, but the increasing cutoffs are more effective.

Threads == 2^d(Depth of recursion)