Run Hou 001090054

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 5**

**Task (List down the tasks performed in the Assignment)**

**Relationship Conclusion: (For ex : z = a \* b)**

**Evidence to support the conclusion:**

**1. 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.**

**Task 1**

Reasonable cut off value

When chosing the same parallelism with 8.

The length of Array is 1,500,000. When cut off value is 190,000, there is a significant improvement of this algorithm.

The length of Array is 2,000,000. When cut off value is 260,000, there is a significant improvement of this algorithm.

The length of Array is 2,500,000. When cut off value is 320,000, there is a significant improvement of this algorithm.

The length of Array is 3,000,000. When cut off value is 390,000, there is a significant improvement of this algorithm.

The length of Array is 3,500,000. When cut off value is 440,000, there is a significant improvement of this algorithm.

The length of Array is 4,000,000. When cut off value is 510,000, there is a significant improvement of this algorithm.

The length of Array is 4,500,000. When cut off value is 570,000, there is a significant improvement of this algorithm.

The length of Array is 5,000,000. When cut off value is 630,000, there is a significant improvement of this algorithm.

The length of Array is 5,500,000. When cut off value is 6,90,000, there is a significant improvement of this algorithm.

图形用户界面, 文本

描述已自动生成

电脑屏幕的照片上有文字

中度可信度描述已自动生成

电脑屏幕截图

中度可信度描述已自动生成

图形用户界面, 文本

描述已自动生成

文本

低可信度描述已自动生成

文本, 日历

描述已自动生成

图形用户界面, 文本

描述已自动生成

图表

描述已自动生成

From the chart we know that cut off value is proportional to the length of array.

Cut\_Of\_Value = k \* Input\_Length;

**Task 2**

When controlling the length of array(2000000), but changing the parallelism.

Parallelism is 2.

图形用户界面, 文本

描述已自动生成

Parallelism is 4.

电脑屏幕的照片上有文字

中度可信度描述已自动生成

Parallelism is 8.

图形用户界面, 文本

中度可信度描述已自动生成

Parallelism is 16.

图形用户界面, 文本

描述已自动生成

图形用户界面, 文本

描述已自动生成

When parallelism is 16, cut off value is larger than 510000,and running time even increases.

Parallelism is 32.

图形用户界面, 文本

描述已自动生成

Result is same as 16.

According to the all the results, parallelism is not the larger the better to improve the performance.

Task 3

From task 1 and task 2, I would like to combine them together. Cut off value is in specific area, which is (0.13 – 0.25) \* array.length.

Parallelism can be determined by experiment, when larger than some specific number, we do not need to use more parallelisms.