Xingyu Huang (001582498)

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 4**

* **Task (List down the tasks performed in the Assignment)**
* **Relationship Conclusion: (For ex : z = a \* b)**

**Conclusion：**

**1.cutoff:**

Obviously, using Parallel Sorting will be much faster than using system sorting, especially when the array size is very large.

When using cutoff to control the Parallel sort or system sort, there is a similar cutoff threshold for arrays of different sizes. In my experiment, this threshold is: when the cutoff is close to 60% -80% of array size, the combination of Parallel sort and system sort will have the greatest efficiency.

There is a threshold and not all sorts are sorted by Parallel sort because the use of recursion for Parallel sorting consumes lots of memory space, it will influence the sort efficiency, so it is necessary to adjust the two sorts in appropriate percent.

2.parallelism degree

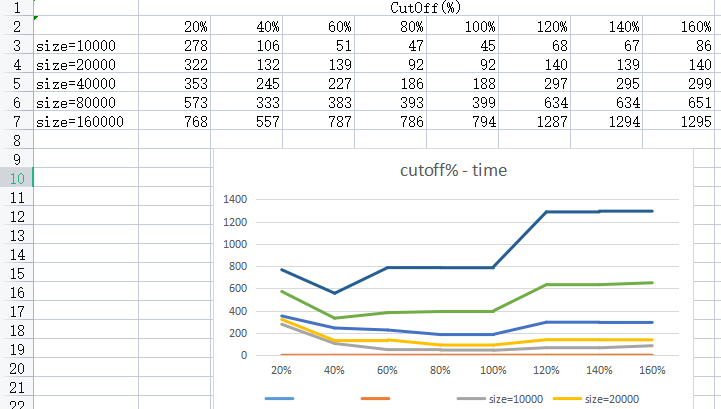
Using parallel threads for sorting is obviously much more efficient than single-threaded sorting (the idea of map-reduce), but there is usually a parallel threshold, that is, after reaching this threshold, no matter how many threads are added for work, or how large of array size. it will not increase efficiency. In my experiment, the thread threshold is 8-10

**Evidence to support the conclusion:**

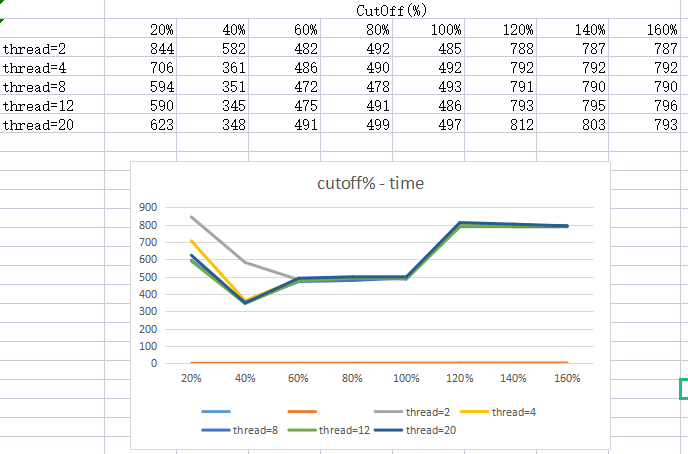
**Output (Snapshot of Code output in the terminal)**

1. **Graphical Representation(Observations from experiments should be tabulated and analyzed by plotting graphs(usually in excel) to arrive on the ßrelationship conclusion)**

Thread numbers = 10 ,with different array size:



Array size = 100000, with different thread numbers



**This main function addr is：**

INFO6205-Fall2021\src\main\java\edu\neu\coe\info6205\sort\par\Main.java

INFO6205-Fall2021\src\main\java\edu\neu\coe\info6205\sort\par\ParSort.java

**Unit tests result:(Snapshot of successful unit tesst run)**

3.Your ****code**** (*RandomWalk.java* plus anything else that you changed or created);

https://github.com/XingyuHuang23/INFO\_6205\_Assignments