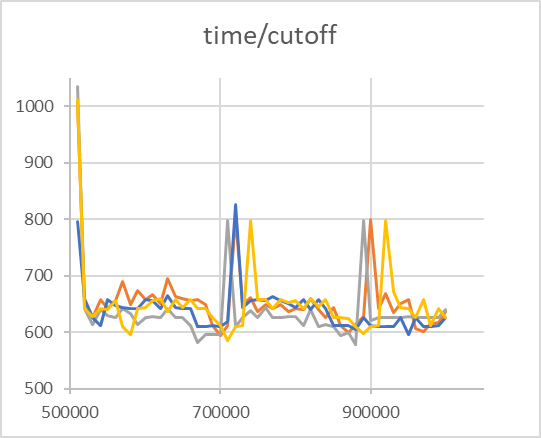
Wenzhuo Wang (1562172)

**Program Structures & Algorithms**

**Fall 2021**

**Assignment No. 5**

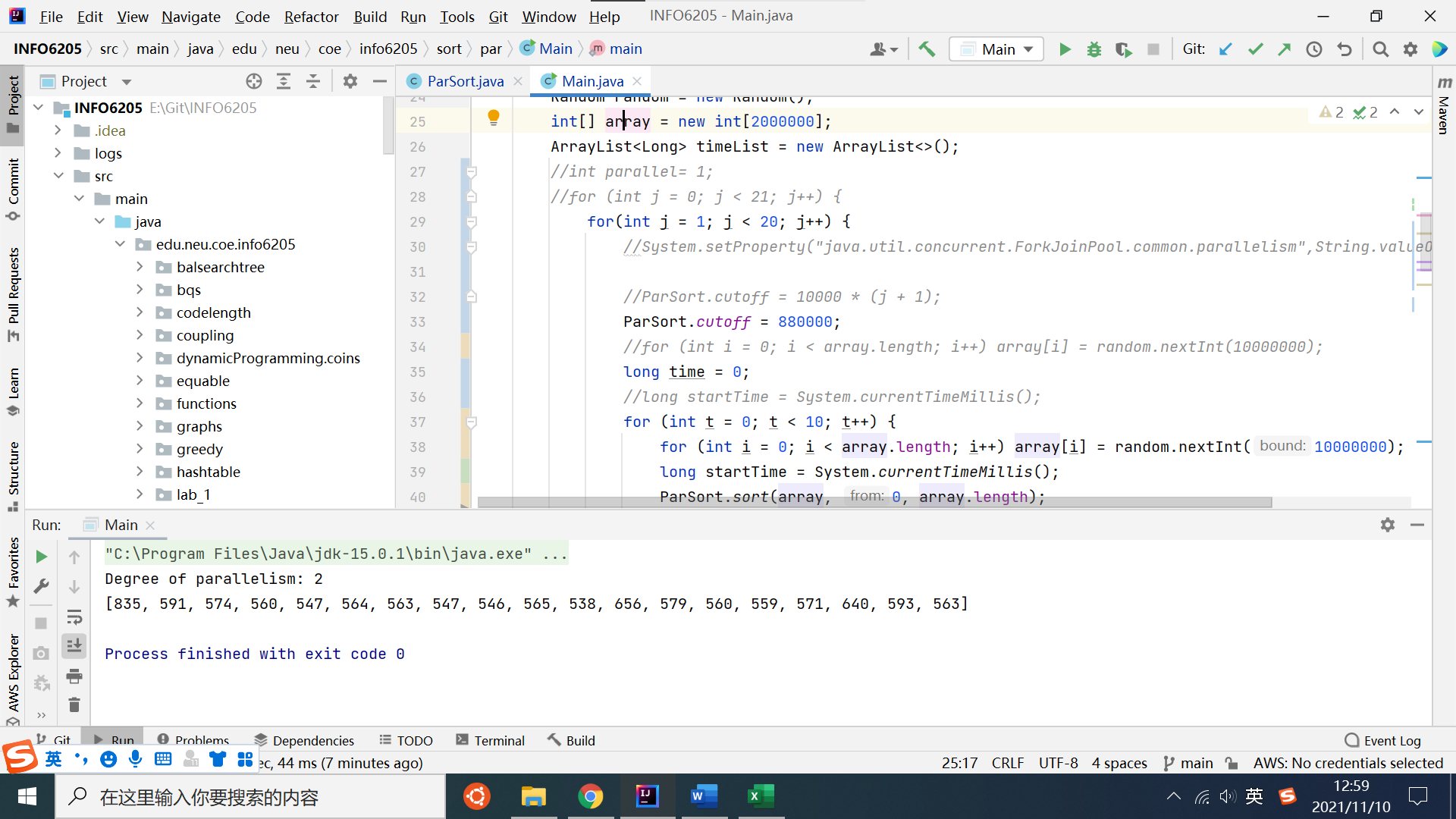
* **Task**
* (Part 1) experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then use the system sort instead.
* (Part 2) decide on an ideal number (t) of separate threads (stick to powers of 2) and arrange for that number of partitions to be parallelized (by preventing recursion after the depth of lg t is reached).
* (Part 3) An appropriate combination of these.
* **Relationship Conclusion:**
* The best cutoff value is 880000.
* The total time consumption varies with the cutoff value, as shown in the figure below. (test 4 times)
* 
* It can be seen that when the cutoff value is about 700000 or 900000, the total time increases sharply to form two peaks. The total time consumption decreases slowly in the range of 500000 to 700000, and so does 700000 to 900000 (possibly due to the warm up time of the system, the previous sorting takes a long time and cannot be analyzed). The best cutoff value should be selected near the left side of these two peaks.
* In this system, the optimal threadcount value is 4(2^2).
* This is the graph of logarithm and time of threadcount average, as shown in the figure below. During the experiment, the cutoff value is set to 880000, the sorting under each thread value is performed 20 times, and the average value of the last 12 times is taken to avoid the impact of the system "warm up".
* As shown in the figure, when the threadcount value is set to 4, that is, the average time for parsort to use 4 threads each time is the shortest.

When threadcount is greater than 4, the time consumption increases linearly with the logarithm of threadcount.

Summary: for an unordered array with a length of 2000000, the optimal cutoff value is about 880000. The best threadcount value for this computer is 4

* **Evidence to support the conclusion:**

1. **Output**

****

* The original data are saved in file assignment5.xlsx

**Graphical Representation**

**Part1:**

**Part2:**