

**Barla Venkat Sandeep (002135635)**

**Program Structures & Algorithms**

**INFO6205**

## **Assignment 5**

### **Task:**

- To implement a parallel sorting algorithm such that each partition of the array is sorted in parallel.

### **Findings:**

For array size = 20000

<b>Cutoff value</b>	<b>Average time for 10 runs(milliseconds)</b>
200	228
300	41
400	77
500	60
600	48
700	49
800	10
900	13
1000	8
1100	19
1200	20
1300	15

1400	17
1500	14
1600	8
1700	7
1800	8
1900	8
2000	7

For array size = 40000

Cutoff value	Average time for 10 runs(milliseconds)
200	342
300	186
400	171
500	28
600	60
700	15
800	19
900	14
1000	17
1100	15
1200	25
1300	16
1400	18
1500	14
1600	17
1700	23
1800	19
1900	12

2000	13
------	----

For array size = 80000

Cutoff value	Average time for 10 runs(milliseconds)
200	650
300	803
400	145
500	61
600	43
700	48
800	30
900	41
1000	30
1100	45
1200	33
1300	33
1400	28
1500	37
1600	33
1700	26
1800	34
1900	24
2000	23

For array size = 160000

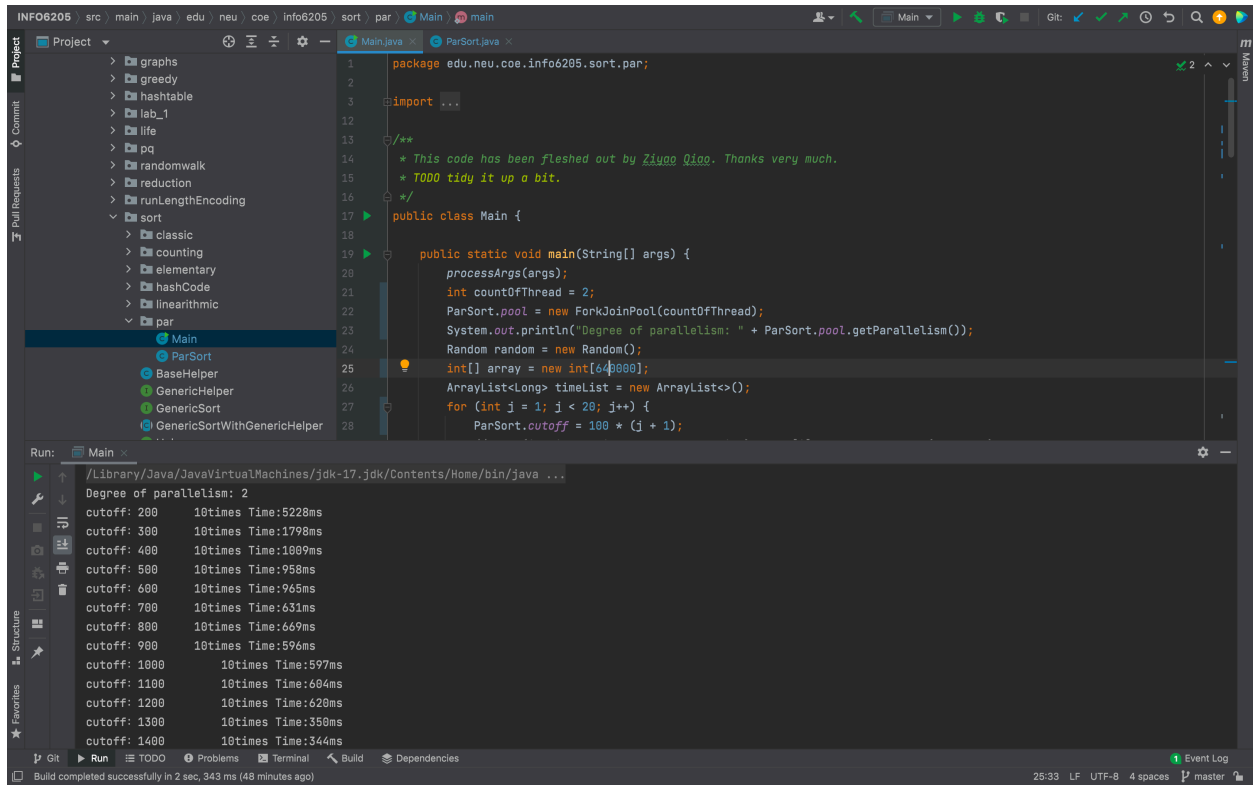
<b>Cutoff value</b>	<b>Average time for 10 runs(milliseconds)</b>
200	1129
300	672
400	140
500	103
600	104
700	73
800	74
900	82
1000	66
1100	69
1200	86
1300	61
1400	51
1500	54
1600	57
1700	51
1800	56
1900	52
2000	58

For array size = 640000

<b>Cutoff value</b>	<b>Average time for 10 runs(milliseconds)</b>
200	5228
300	1798

400	1009
500	958
600	965
700	631
800	669
900	596
1000	597
1100	604
1200	620
1300	350
1400	344
1500	344
1600	377
1700	338
1800	398
1900	331
2000	336

# Output:



The screenshot displays an IDE with a project named 'INFO6205'. The project structure on the left includes folders like 'graphs', 'greedy', 'hashtable', 'lab\_1', 'life', 'pq', 'randomwalk', 'reduction', 'runLengthEncoding', 'sort', and 'par'. The 'par' folder is expanded, showing 'Main' and 'ParSort' files. The 'Main' file is open in the editor, showing the following code:

```
1 package edu.neu.coe.info6205.sort.par;
2
3 import ...
4
5 /**
6  * This code has been fleshed out by Ziyao Qiao. Thanks very much.
7  * TODO tidy it up a bit.
8  */
9
10 public class Main {
11
12     public static void main(String[] args) {
13         processArgs(args);
14         int countOfThread = 2;
15         ParSort.pool = new ForkJoinPool(countOfThread);
16         System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism());
17         Random random = new Random();
18         int[] array = new int[40000];
19         ArrayList<Long> timeList = new ArrayList<>();
20         for (int j = 1; j < 20; j++) {
21             ParSort.cutoff = 100 * (j + 1);
22         }
23     }
24 }
```

The 'Run' console at the bottom shows the output of the program:

```
/Library/Java/JavaVirtualMachines/jdk-17.jdk/Contents/Home/bin/java ...
Degree of parallelism: 2
cutoff: 200      10times Time:5228ms
cutoff: 300      10times Time:1798ms
cutoff: 400      10times Time:1009ms
cutoff: 500      10times Time:958ms
cutoff: 600      10times Time:965ms
cutoff: 700      10times Time:631ms
cutoff: 800      10times Time:669ms
cutoff: 900      10times Time:596ms
cutoff: 1000     10times Time:597ms
cutoff: 1100     10times Time:604ms
cutoff: 1200     10times Time:628ms
cutoff: 1300     10times Time:350ms
cutoff: 1400     10times Time:344ms
```

The status bar at the bottom indicates 'Build completed successfully in 2 sec, 343 ms (48 minutes ago)' and '25:33 LF UTF-8 4 spaces master'.