XINHUI ZHAO (001560851)

Program Structures & Algorithms

Fall 2021

Assignment No. 5

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

Your 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.

1. 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.
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.

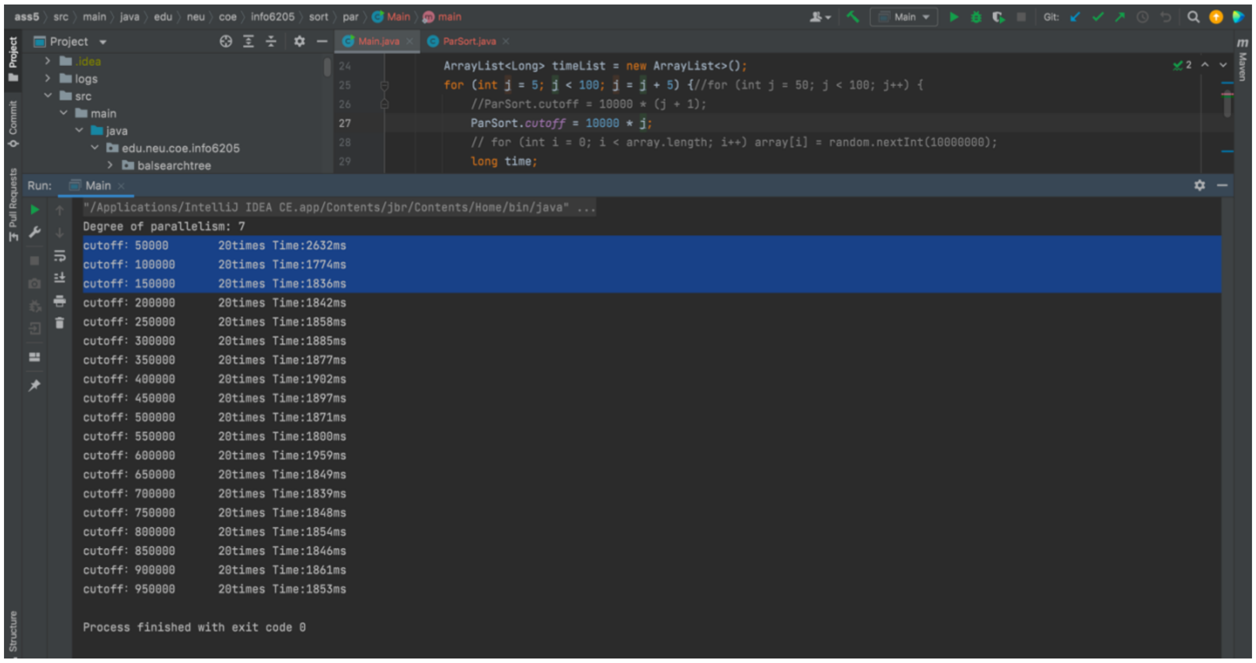
You must prepare a report that shows the results of your experiments and draws a conclusion (or more) about the efficacy of this method of parallelizing sort. Your experiments should involve sorting arrays of sufficient size for the parallel sort to make a difference. You should run with many different array sizes (they must be sufficiently large to make parallel sorting worthwhile, obviously) and different cutoff schemes.

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

Cutoff: 100,000
Threads number: 4

⊙ Evidence to support the conclusion:

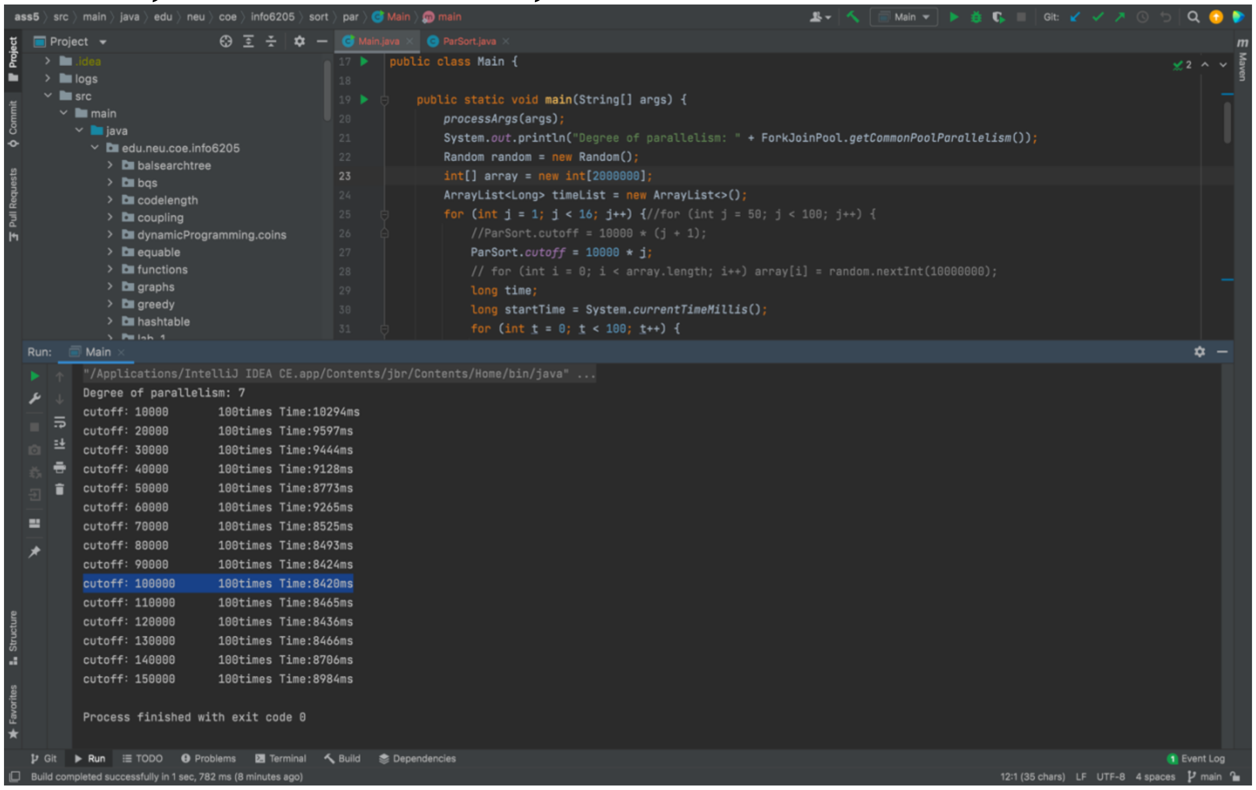
1. Output (Snapshot of Code output in the terminal)



```
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 5; j < 100; j = j + 5) {//for (int j = 50; j < 100; j++) {
            //ParSort.cutoff = 10000 * (j + 1);
            ParSort.cutoff = 10000 * j;
            // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
            long time;
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff: 50000       20times Time:2632ms
cutoff: 100000      20times Time:1774ms
cutoff: 150000      20times Time:1836ms
cutoff: 200000      20times Time:1842ms
cutoff: 250000      20times Time:1858ms
cutoff: 300000      20times Time:1885ms
cutoff: 350000      20times Time:1877ms
cutoff: 400000      20times Time:1902ms
cutoff: 450000      20times Time:1897ms
cutoff: 500000      20times Time:1871ms
cutoff: 550000      20times Time:1800ms
cutoff: 600000      20times Time:1959ms
cutoff: 650000      20times Time:1849ms
cutoff: 700000      20times Time:1839ms
cutoff: 750000      20times Time:1848ms
cutoff: 800000      20times Time:1854ms
cutoff: 850000      20times Time:1846ms
cutoff: 900000      20times Time:1861ms
cutoff: 950000      20times Time:1853ms

Process finished with exit code 0
```

When the array.length is 2,000,000, using the default ForkJoinPool and then running the cutoff from 50,000 to 950,000, it shows that the best case is around 100,000. So I changed the times to 100 and running the cutoff from 10,000 to 150,000, and actually, it shows that 100,000 is the best cutoff, as below.

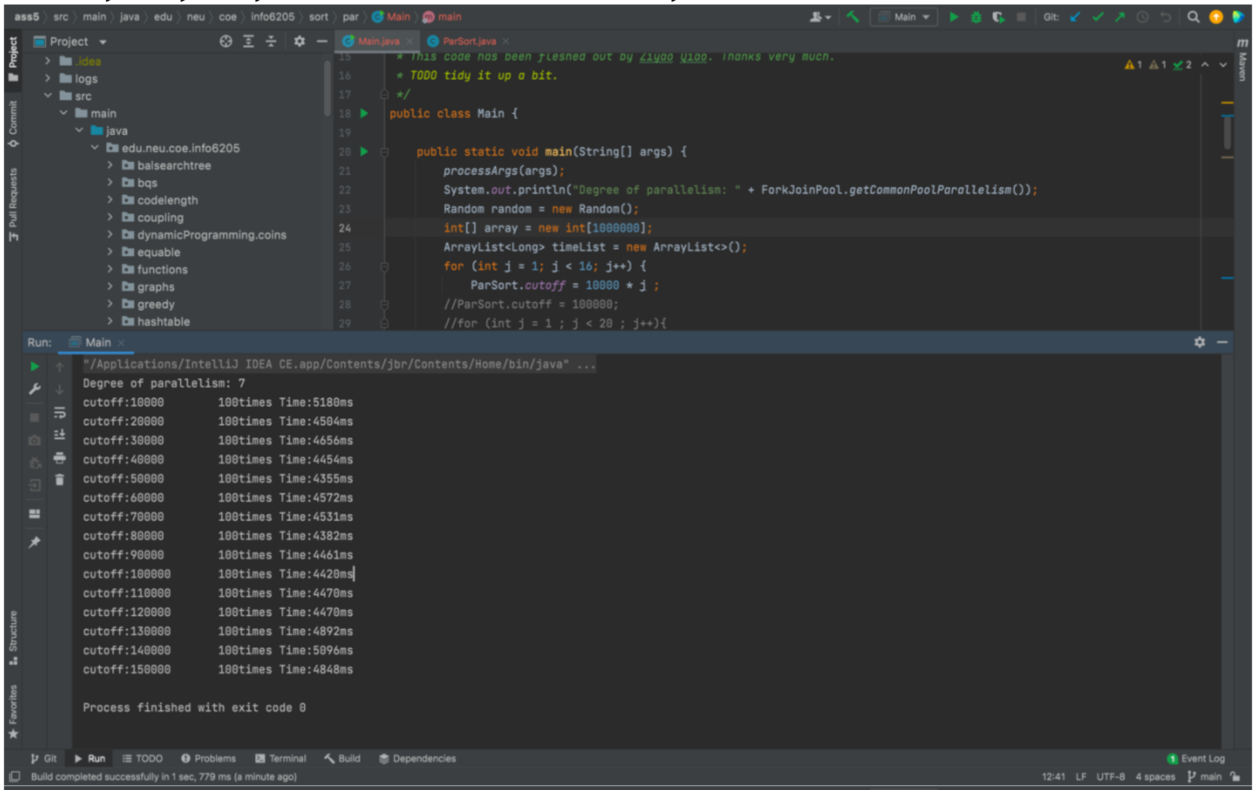


```
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[2000000];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j < 16; j++) {//for (int j = 50; j < 100; j++) {
            //ParSort.cutoff = 10000 * (j + 1);
            ParSort.cutoff = 10000 * j;
            // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
            long time;
            long startTime = System.currentTimeMillis();
            for (int t = 0; t < 100; t++) {
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff: 10000       100times Time:10294ms
cutoff: 20000       100times Time:9597ms
cutoff: 30000       100times Time:9444ms
cutoff: 40000       100times Time:9128ms
cutoff: 50000       100times Time:8773ms
cutoff: 60000       100times Time:9265ms
cutoff: 70000       100times Time:8525ms
cutoff: 80000       100times Time:8493ms
cutoff: 90000       100times Time:8424ms
cutoff: 100000      100times Time:8420ms
cutoff: 110000      100times Time:8465ms
cutoff: 120000      100times Time:8436ms
cutoff: 130000      100times Time:8466ms
cutoff: 140000      100times Time:8706ms
cutoff: 150000      100times Time:8984ms

Process finished with exit code 0
```

Build completed successfully in 1 sec, 782 ms (8 minutes ago)

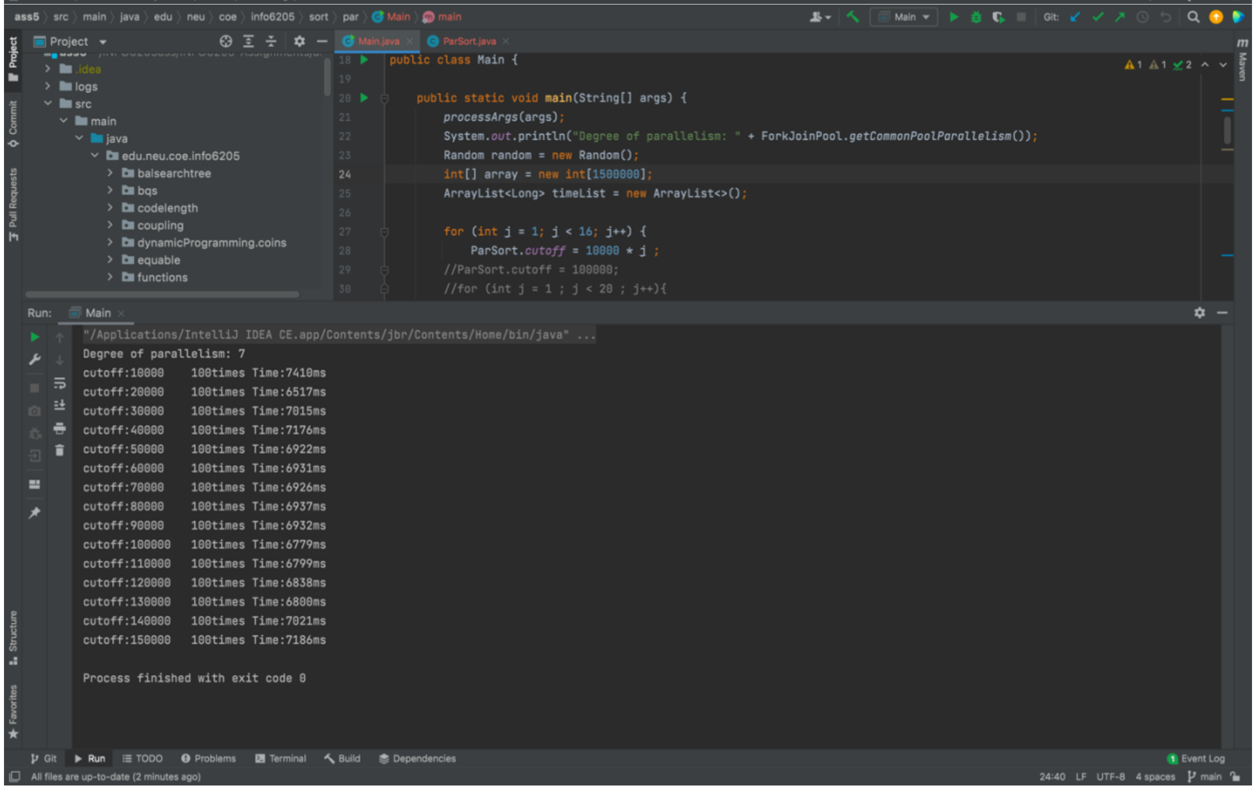
And I changed the array.length to 1,000,000, 1,500,000, 2,500,000 and 3,000,000, it also seems that 100,000 is the best value of cutoff.



```java
     * This code has been fleshed out by Ziyao Qiao. Thanks very much.
     * TODO tidy it up a bit.
     */
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[1000000];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
        //ParSort.cutoff = 100000;
        //for (int j = 1 ; j < 20 ; j++){
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:10000        100times Time:5180ms
cutoff:20000        100times Time:4504ms
cutoff:30000        100times Time:4656ms
cutoff:40000        100times Time:4454ms
cutoff:50000        100times Time:4355ms
cutoff:60000        100times Time:4572ms
cutoff:70000        100times Time:4531ms
cutoff:80000        100times Time:4382ms
cutoff:90000        100times Time:4461ms
cutoff:100000       100times Time:4420ms
cutoff:110000       100times Time:4470ms
cutoff:120000       100times Time:4470ms
cutoff:130000       100times Time:4892ms
cutoff:140000       100times Time:5096ms
cutoff:150000       100times Time:4848ms

Process finished with exit code 0
```

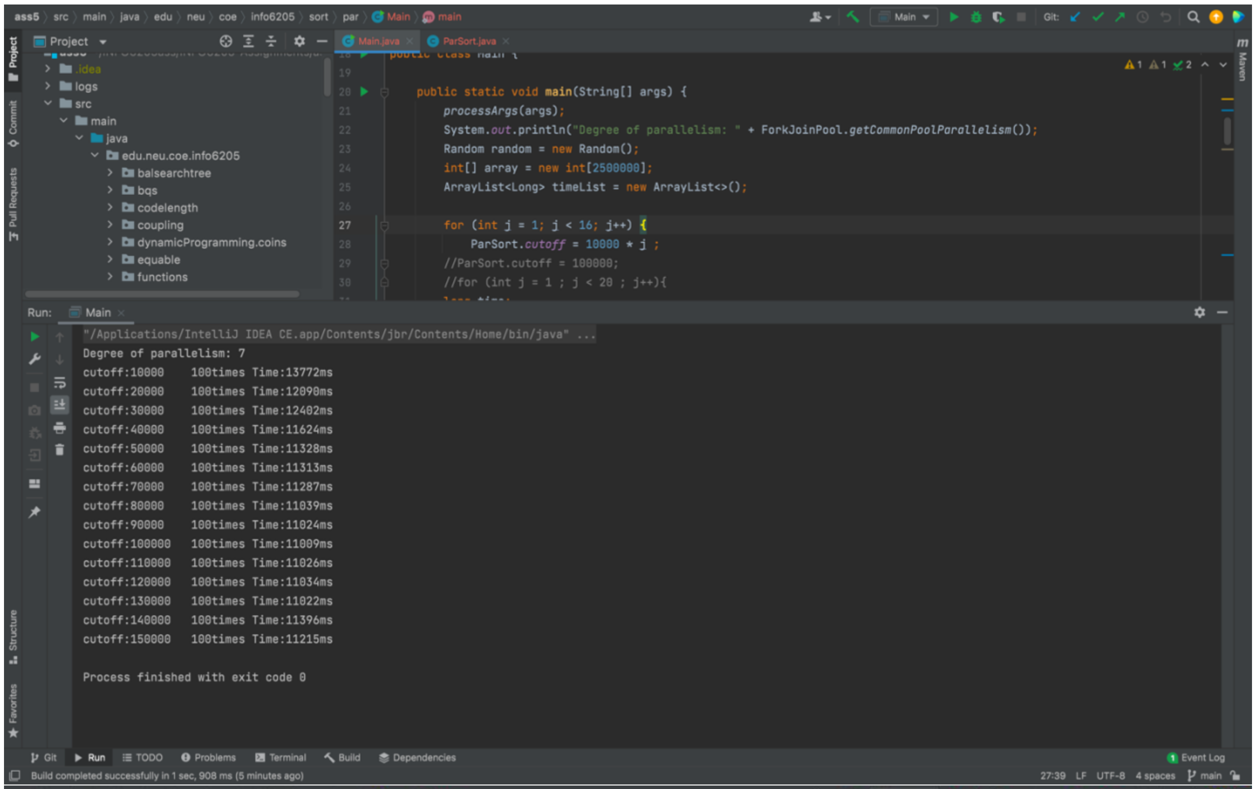
Build completed successfully in 1 sec, 779 ms (a minute ago)



```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[1500000];
        ArrayList<Long> timeList = new ArrayList<>();

        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
        //ParSort.cutoff = 100000;
        //for (int j = 1 ; j < 20 ; j++){
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:10000    100times Time:7410ms
cutoff:20000    100times Time:6517ms
cutoff:30000    100times Time:7015ms
cutoff:40000    100times Time:7176ms
cutoff:50000    100times Time:6922ms
cutoff:60000    100times Time:6931ms
cutoff:70000    100times Time:6926ms
cutoff:80000    100times Time:6937ms
cutoff:90000    100times Time:6932ms
cutoff:100000   100times Time:6779ms
cutoff:110000   100times Time:6799ms
cutoff:120000   100times Time:6838ms
cutoff:130000   100times Time:6800ms
cutoff:140000   100times Time:7021ms
cutoff:150000   100times Time:7186ms

Process finished with exit code 0
```

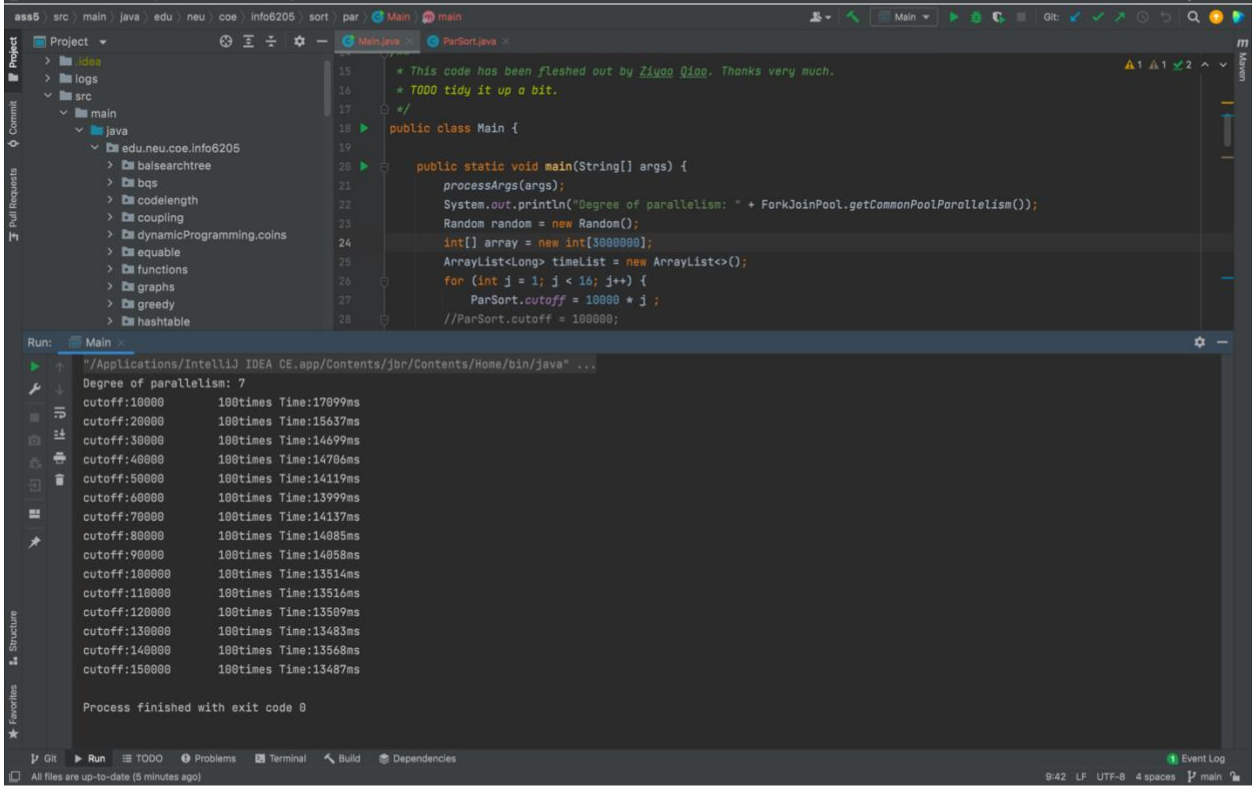
All files are up-to-date (2 minutes ago)



```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[2500000];
        ArrayList<Long> timeList = new ArrayList<>();

        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
        //ParSort.cutoff = 100000;
        //for (int j = 1 ; j < 20 ; j++){
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:10000    100times Time:13772ms
cutoff:20000    100times Time:12090ms
cutoff:30000    100times Time:12402ms
cutoff:40000    100times Time:11624ms
cutoff:50000    100times Time:11328ms
cutoff:60000    100times Time:11313ms
cutoff:70000    100times Time:11287ms
cutoff:80000    100times Time:11039ms
cutoff:90000    100times Time:11024ms
cutoff:100000   100times Time:11009ms
cutoff:110000   100times Time:11026ms
cutoff:120000   100times Time:11034ms
cutoff:130000   100times Time:11022ms
cutoff:140000   100times Time:11396ms
cutoff:150000   100times Time:11215ms

Process finished with exit code 0
```

Build completed successfully in 1 sec, 908 ms (5 minutes ago)

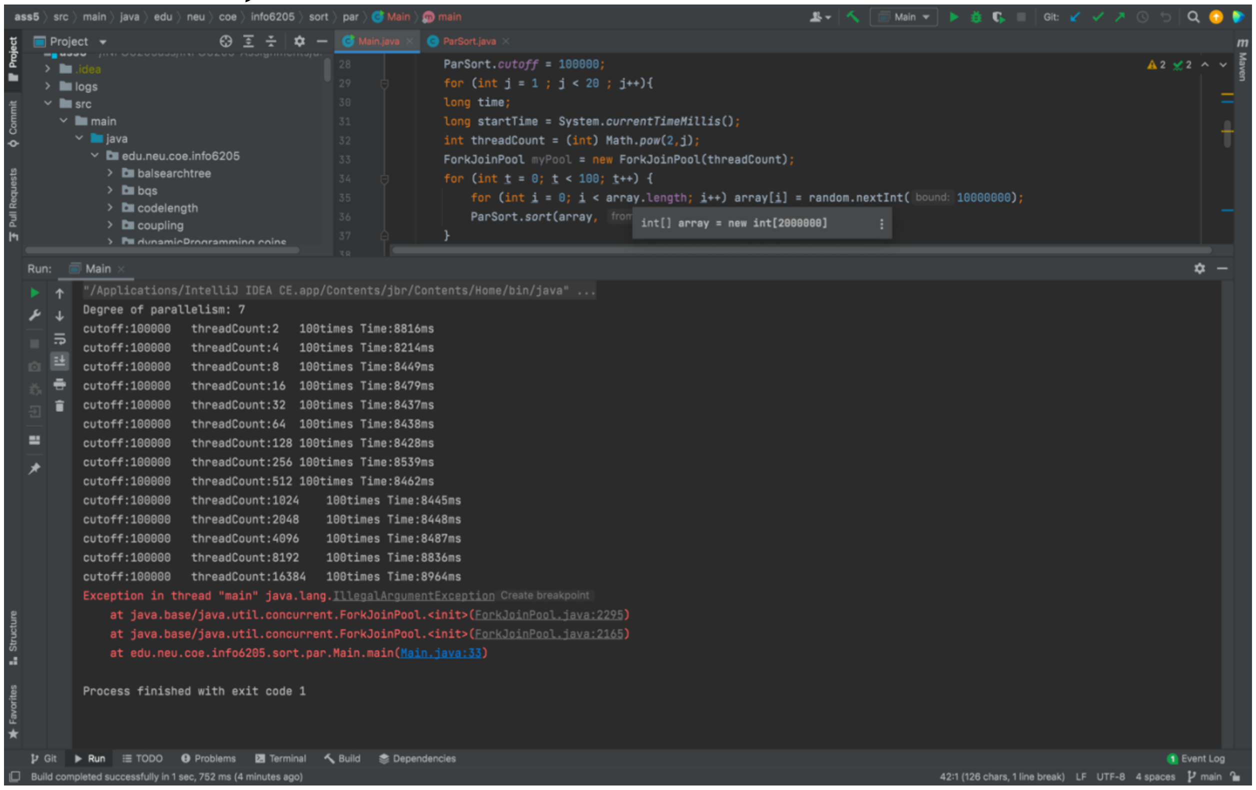


```java
 * This code has been fleshed out by Ziyao Qiao. Thanks very much.
 * TODO tidy it up a bit.
 */
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[3000000];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
        //ParSort.cutoff = 100000;
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:10000        100times Time:17099ms
cutoff:20000        100times Time:15637ms
cutoff:30000        100times Time:14699ms
cutoff:40000        100times Time:14706ms
cutoff:50000        100times Time:14119ms
cutoff:60000        100times Time:13999ms
cutoff:70000        100times Time:14137ms
cutoff:80000        100times Time:14085ms
cutoff:90000        100times Time:14058ms
cutoff:100000       100times Time:13514ms
cutoff:110000       100times Time:13516ms
cutoff:120000       100times Time:13509ms
cutoff:130000       100times Time:13483ms
cutoff:140000       100times Time:13568ms
cutoff:150000       100times Time:13487ms

Process finished with exit code 0
```

All files are up-to-date (5 minutes ago)

When the cutoff is 100,000 and array.length is 2,000,000, running the thread count from 2^1 to 2^20, and the max value of thread count is 2^14, We can find the best value of thread count is 4.



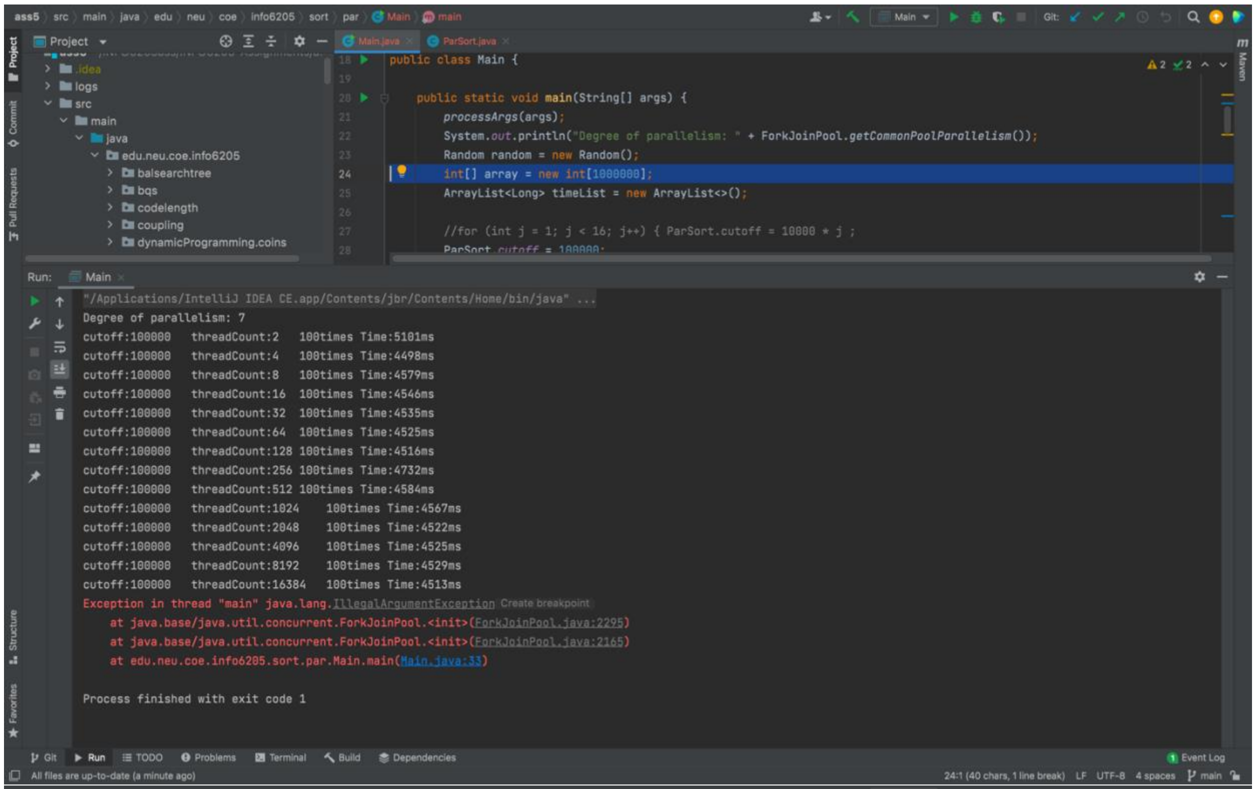
```java
28        ParSort.cutoff = 100000;
29        for (int j = 1 ; j < 20 ; j++){
30        long time;
31        long startTime = System.currentTimeMillis();
32        int threadCount = (int) Math.pow(2,j);
33        ForkJoinPool myPool = new ForkJoinPool(threadCount);
34        for (int t = 0; t < 100; t++) {
35            for (int i = 0; i < array.length; i++) array[i] = random.nextInt( bound: 10000000);
36            ParSort.sort(array,  from
37        }
```

```
int[] array = new int[2000000]
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:100000   threadCount:2   100times Time:8816ms
cutoff:100000   threadCount:4   100times Time:8214ms
cutoff:100000   threadCount:8   100times Time:8449ms
cutoff:100000   threadCount:16  100times Time:8479ms
cutoff:100000   threadCount:32  100times Time:8437ms
cutoff:100000   threadCount:64  100times Time:8438ms
cutoff:100000   threadCount:128 100times Time:8428ms
cutoff:100000   threadCount:256 100times Time:8539ms
cutoff:100000   threadCount:512 100times Time:8462ms
cutoff:100000   threadCount:1024    100times Time:8445ms
cutoff:100000   threadCount:2048    100times Time:8448ms
cutoff:100000   threadCount:4096    100times Time:8487ms
cutoff:100000   threadCount:8192    100times Time:8836ms
cutoff:100000   threadCount:16384   100times Time:8964ms
Exception in thread "main" java.lang.IllegalArgumentException Create breakpoint
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2295)
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2165)
    at edu.neu.coe.info6205.sort.par.Main.main(Main.java:33)

Process finished with exit code 1
```

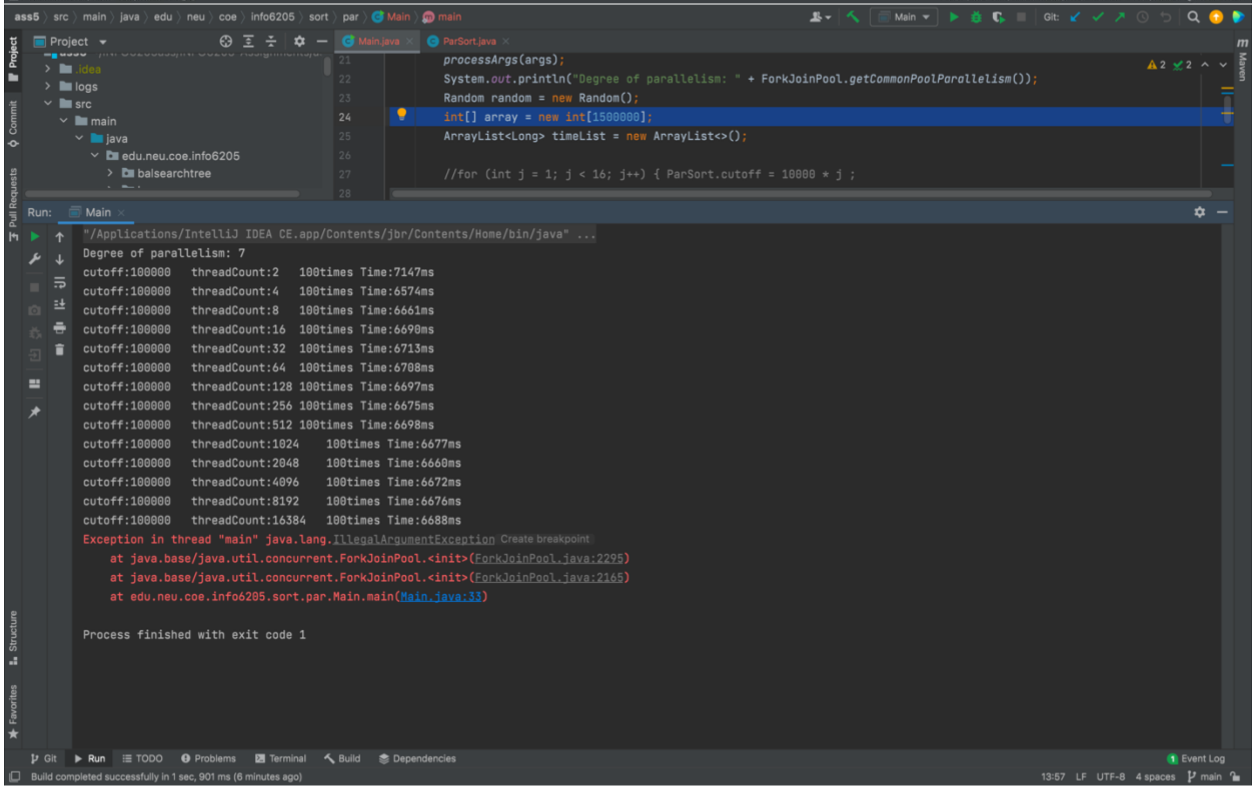
Build completed successfully in 1 sec, 752 ms (4 minutes ago)

It comes the same results when changed the array.list to 1,000,000, 1,500,000, 2,500,000 and 3,000,000, as below.


ass5 › src › main › java › edu › neu › coe › info6205 › sort › par › Main › main
Project
Main.java
ParSort.java
.idea
logs
src
main
java
edu.neu.coe.info6205
balsearchtree
bqs
codelength
coupling
dynamicProgramming.coins
Run: Main
Git Run TODO Problems Terminal Build Dependencies
Event Log
All files are up-to-date (a minute ago)
24:1 (40 chars, 1 line break) LF UTF-8 4 spaces main

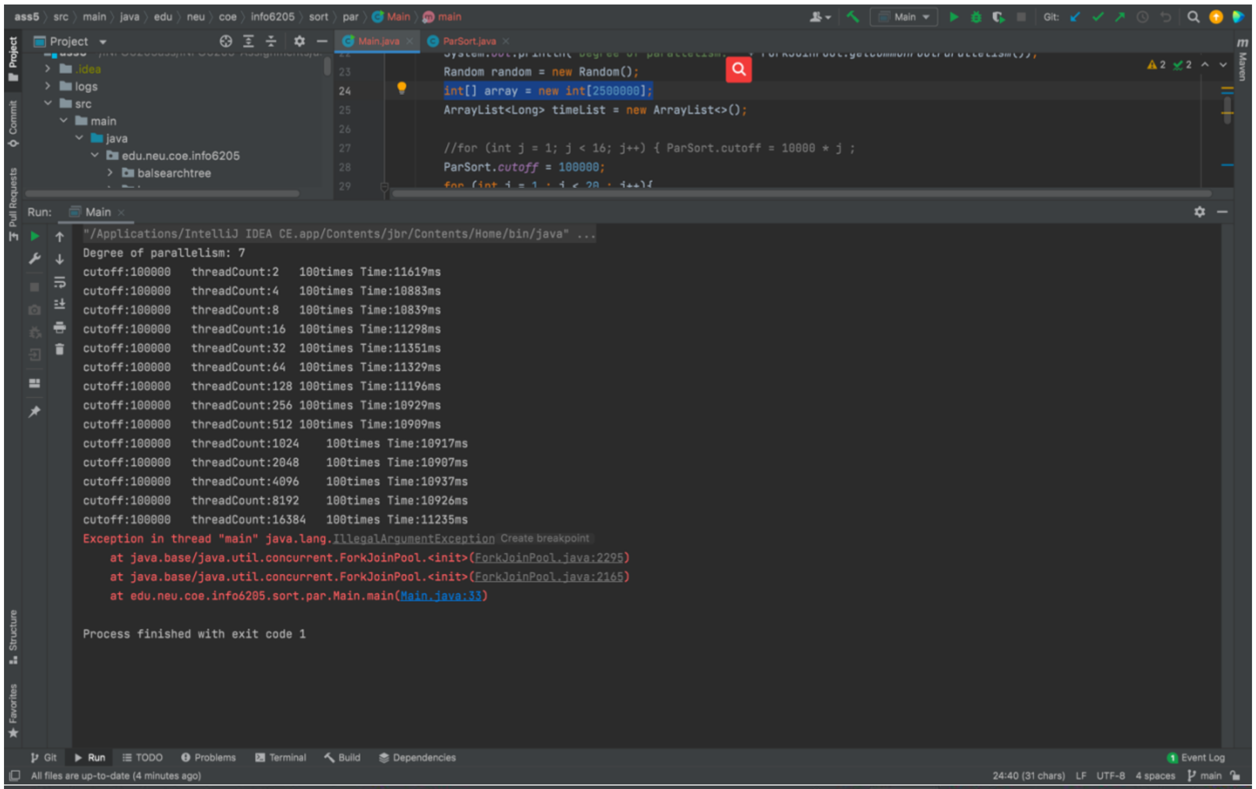
```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[1000000];
        ArrayList<Long> timeList = new ArrayList<>();

        //for (int j = 1; j < 16; j++) { ParSort.cutoff = 10000 * j ;
        ParSort.cutoff = 100000;
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:100000   threadCount:2   100times Time:5101ms
cutoff:100000   threadCount:4   100times Time:4498ms
cutoff:100000   threadCount:8   100times Time:4579ms
cutoff:100000   threadCount:16  100times Time:4546ms
cutoff:100000   threadCount:32  100times Time:4535ms
cutoff:100000   threadCount:64  100times Time:4525ms
cutoff:100000   threadCount:128 100times Time:4516ms
cutoff:100000   threadCount:256 100times Time:4732ms
cutoff:100000   threadCount:512 100times Time:4584ms
cutoff:100000   threadCount:1024    100times Time:4567ms
cutoff:100000   threadCount:2048    100times Time:4522ms
cutoff:100000   threadCount:4096    100times Time:4525ms
cutoff:100000   threadCount:8192    100times Time:4529ms
cutoff:100000   threadCount:16384    100times Time:4513ms
Exception in thread "main" java.lang.IllegalArgumentException Create breakpoint
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2295)
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2165)
    at edu.neu.coe.info6205.sort.par.Main.main(Main.java:33)

Process finished with exit code 1
```


ass5 › src › main › java › edu › neu › coe › info6205 › sort › par › Main › main
Project
Main.java
ParSort.java
.idea
logs
src
main
java
edu.neu.coe.info6205
balsearchtree
Run: Main
Git Run TODO Problems Terminal Build Dependencies
Event Log
Build completed successfully in 1 sec, 901 ms (6 minutes ago)
13:57 LF UTF-8 4 spaces main

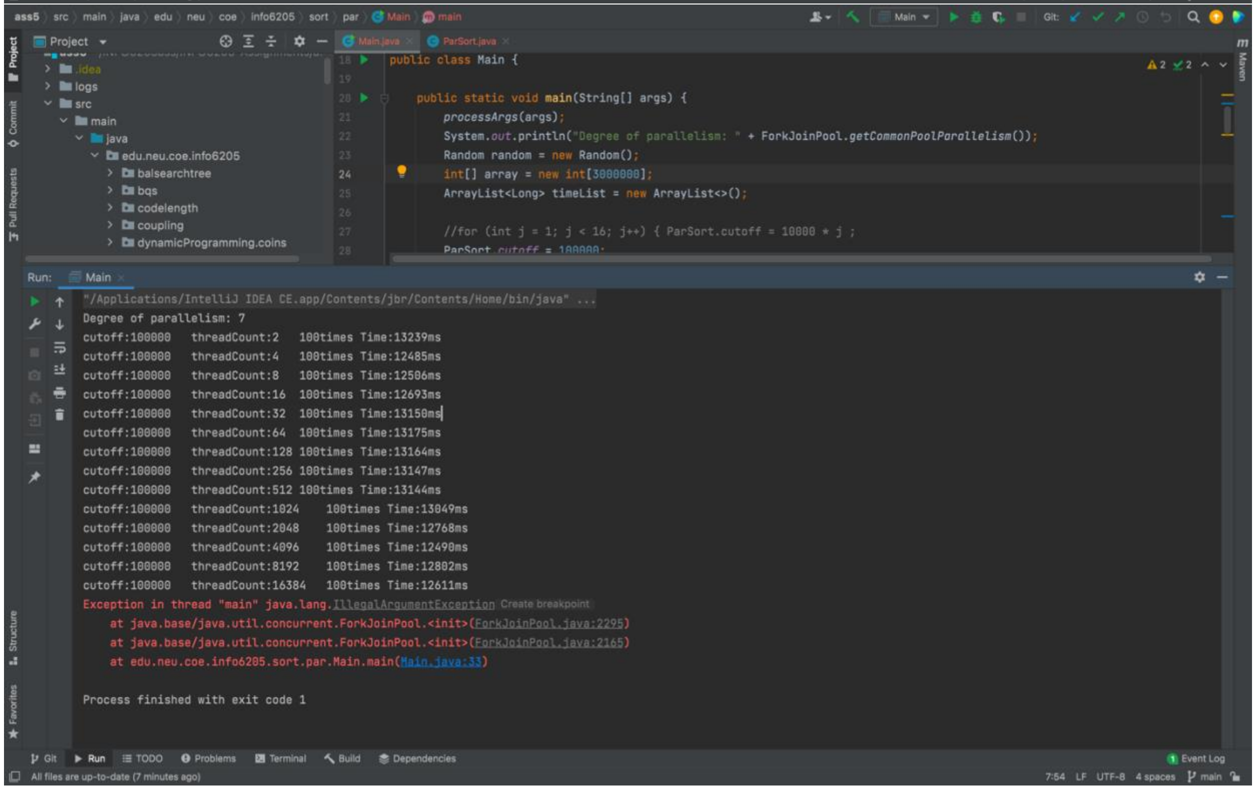
```java
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[1500000];
        ArrayList<Long> timeList = new ArrayList<>();

        //for (int j = 1; j < 16; j++) { ParSort.cutoff = 10000 * j ;
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:100000   threadCount:2   100times Time:7147ms
cutoff:100000   threadCount:4   100times Time:6574ms
cutoff:100000   threadCount:8   100times Time:6661ms
cutoff:100000   threadCount:16  100times Time:6690ms
cutoff:100000   threadCount:32  100times Time:6713ms
cutoff:100000   threadCount:64  100times Time:6708ms
cutoff:100000   threadCount:128 100times Time:6697ms
cutoff:100000   threadCount:256 100times Time:6675ms
cutoff:100000   threadCount:512 100times Time:6698ms
cutoff:100000   threadCount:1024    100times Time:6677ms
cutoff:100000   threadCount:2048    100times Time:6660ms
cutoff:100000   threadCount:4096    100times Time:6672ms
cutoff:100000   threadCount:8192    100times Time:6676ms
cutoff:100000   threadCount:16384    100times Time:6688ms
Exception in thread "main" java.lang.IllegalArgumentException Create breakpoint
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2295)
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2165)
    at edu.neu.coe.info6205.sort.par.Main.main(Main.java:33)

Process finished with exit code 1
```



```java
Random random = new Random();
int[] array = new int[2500000];
ArrayList<Long> timeList = new ArrayList<>();

//for (int j = 1; j < 16; j++) { ParSort.cutoff = 10000 * j ;
ParSort.cutoff = 100000;
```

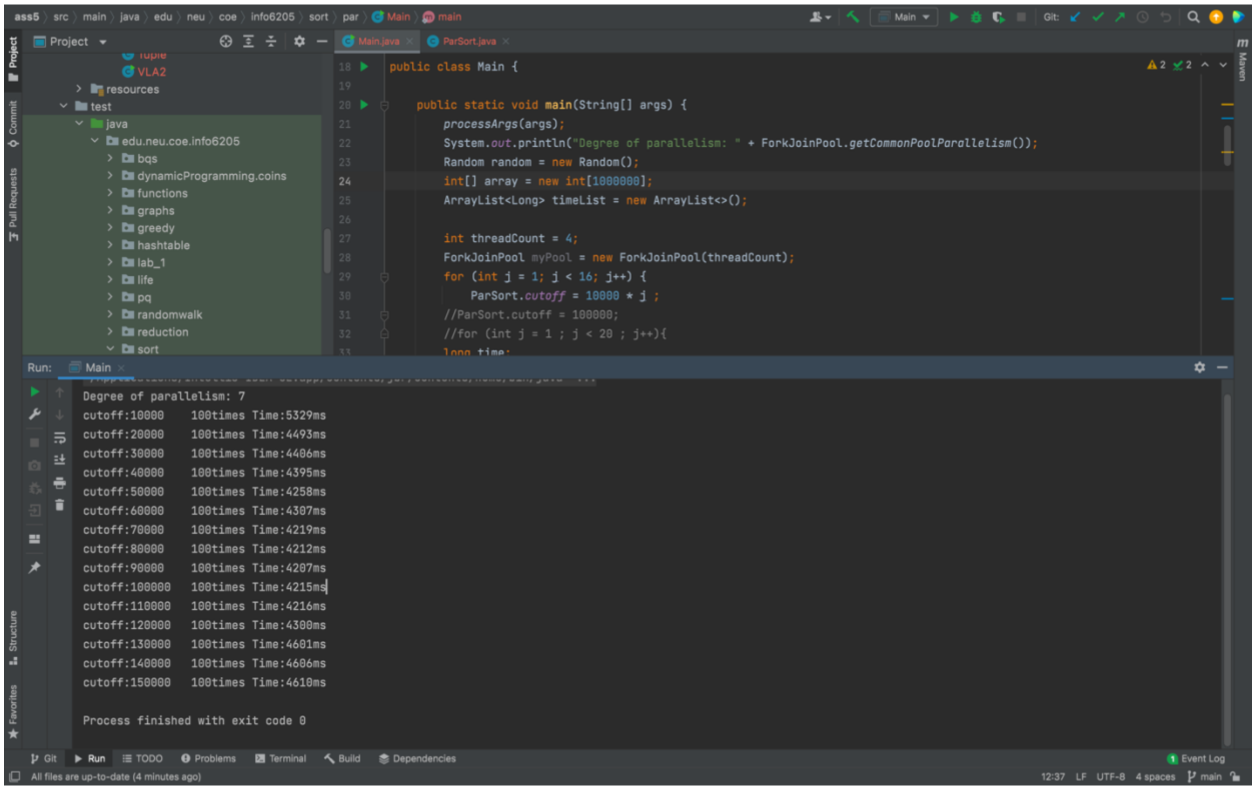
```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:100000   threadCount:2   100times Time:11619ms
cutoff:100000   threadCount:4   100times Time:10883ms
cutoff:100000   threadCount:8   100times Time:10839ms
cutoff:100000   threadCount:16  100times Time:11298ms
cutoff:100000   threadCount:32  100times Time:11351ms
cutoff:100000   threadCount:64  100times Time:11329ms
cutoff:100000   threadCount:128 100times Time:11196ms
cutoff:100000   threadCount:256 100times Time:10929ms
cutoff:100000   threadCount:512 100times Time:10909ms
cutoff:100000   threadCount:1024    100times Time:10917ms
cutoff:100000   threadCount:2048    100times Time:10907ms
cutoff:100000   threadCount:4096    100times Time:10937ms
cutoff:100000   threadCount:8192    100times Time:10926ms
cutoff:100000   threadCount:16384   100times Time:11235ms
Exception in thread "main" java.lang.IllegalArgumentException
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2295)
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2165)
    at edu.neu.coe.info6205.sort.par.Main.main(Main.java:33)

Process finished with exit code 1
```



```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[3000000];
        ArrayList<Long> timeList = new ArrayList<>();

        //for (int j = 1; j < 16; j++) { ParSort.cutoff = 10000 * j ;
```

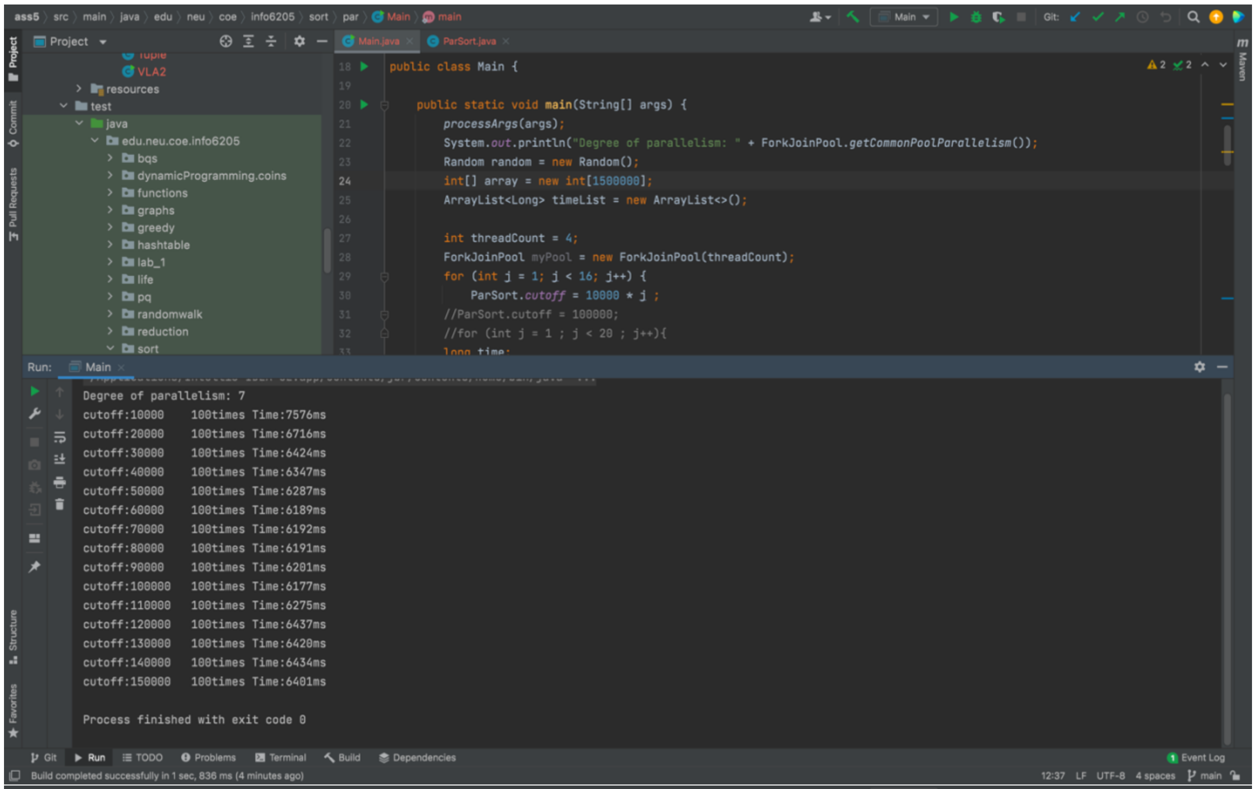
```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:100000   threadCount:2   100times Time:13239ms
cutoff:100000   threadCount:4   100times Time:12485ms
cutoff:100000   threadCount:8   100times Time:12506ms
cutoff:100000   threadCount:16  100times Time:12693ms
cutoff:100000   threadCount:32  100times Time:13150ms
cutoff:100000   threadCount:64  100times Time:13175ms
cutoff:100000   threadCount:128 100times Time:13164ms
cutoff:100000   threadCount:256 100times Time:13147ms
cutoff:100000   threadCount:512 100times Time:13144ms
cutoff:100000   threadCount:1024    100times Time:13049ms
cutoff:100000   threadCount:2048    100times Time:12768ms
cutoff:100000   threadCount:4096    100times Time:12490ms
cutoff:100000   threadCount:8192    100times Time:12802ms
cutoff:100000   threadCount:16384   100times Time:12611ms
Exception in thread "main" java.lang.IllegalArgumentException
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2295)
    at java.base/java.util.concurrent.ForkJoinPool.<init>(ForkJoinPool.java:2165)
    at edu.neu.coe.info6205.sort.par.Main.main(Main.java:33)

Process finished with exit code 1
```

When the thread count is 4, for the array.length 1,000,000, 1,500,000, 2,000,000, 2,500,000 and 3,000,000, running cutoff from 10,000 to 150,000, the best value of cutoff is also around 100,000 as below.



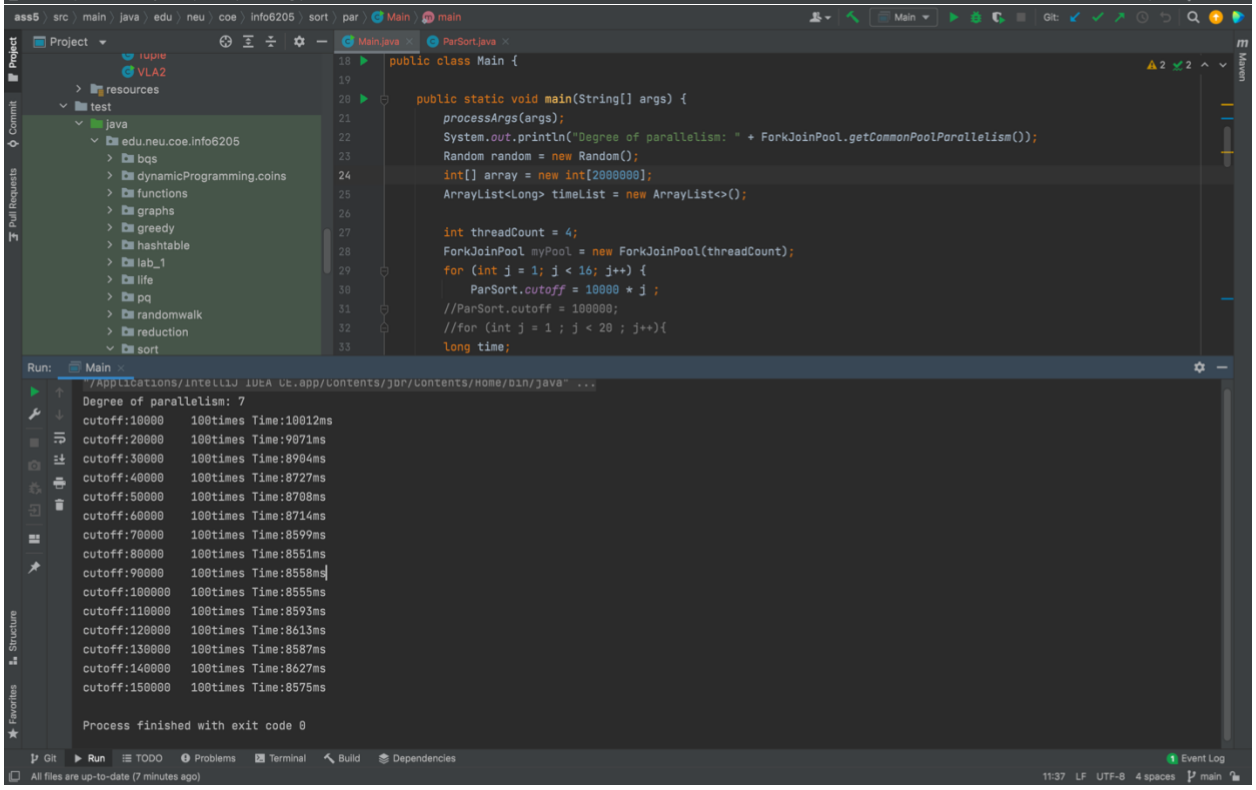
```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[1000000];
        ArrayList<Long> timeList = new ArrayList<>();

        int threadCount = 4;
        ForkJoinPool myPool = new ForkJoinPool(threadCount);
        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
        //ParSort.cutoff = 100000;
        //for (int j = 1 ; j < 20 ; j++){
        long time;
```

```
Degree of parallelism: 7
cutoff:10000    100times Time:5329ms
cutoff:20000    100times Time:4493ms
cutoff:30000    100times Time:4406ms
cutoff:40000    100times Time:4395ms
cutoff:50000    100times Time:4258ms
cutoff:60000    100times Time:4307ms
cutoff:70000    100times Time:4219ms
cutoff:80000    100times Time:4212ms
cutoff:90000    100times Time:4207ms
cutoff:100000   100times Time:4215ms
cutoff:110000   100times Time:4216ms
cutoff:120000   100times Time:4300ms
cutoff:130000   100times Time:4601ms
cutoff:140000   100times Time:4606ms
cutoff:150000   100times Time:4610ms

Process finished with exit code 0
```



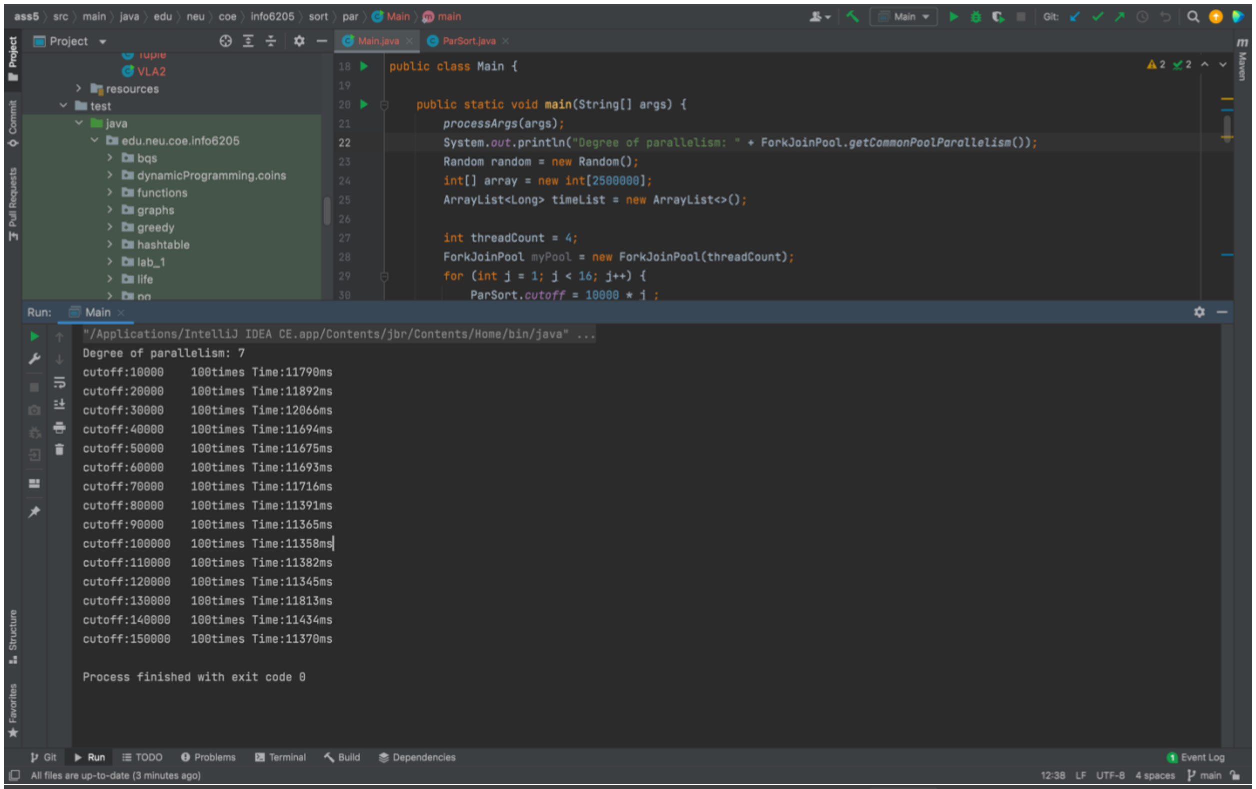
```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[1500000];
        ArrayList<Long> timeList = new ArrayList<>();

        int threadCount = 4;
        ForkJoinPool myPool = new ForkJoinPool(threadCount);
        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
        //ParSort.cutoff = 100000;
        //for (int j = 1 ; j < 20 ; j++){
        long time;
```

```
Degree of parallelism: 7
cutoff:10000    100times Time:7576ms
cutoff:20000    100times Time:6716ms
cutoff:30000    100times Time:6424ms
cutoff:40000    100times Time:6347ms
cutoff:50000    100times Time:6287ms
cutoff:60000    100times Time:6189ms
cutoff:70000    100times Time:6192ms
cutoff:80000    100times Time:6191ms
cutoff:90000    100times Time:6201ms
cutoff:100000   100times Time:6177ms
cutoff:110000   100times Time:6275ms
cutoff:120000   100times Time:6437ms
cutoff:130000   100times Time:6420ms
cutoff:140000   100times Time:6434ms
cutoff:150000   100times Time:6401ms

Process finished with exit code 0
```



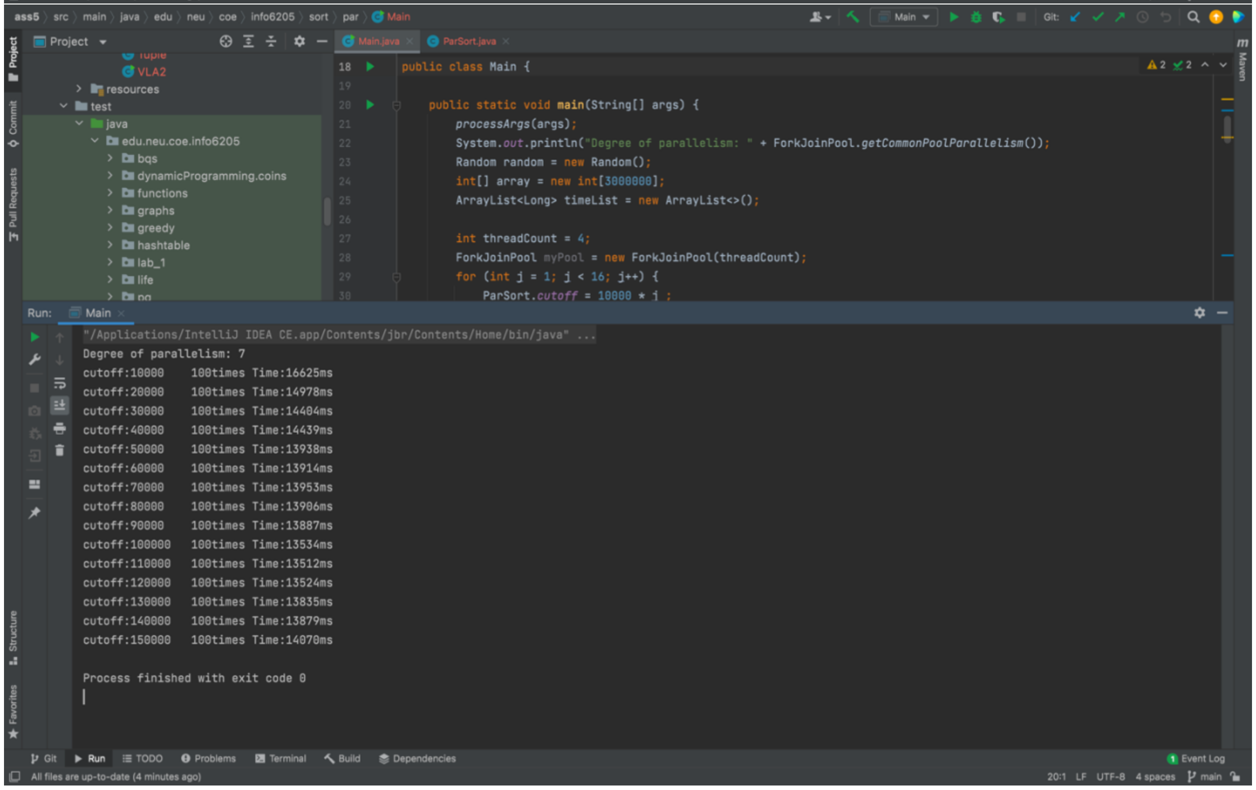
```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[2000000];
        ArrayList<Long> timeList = new ArrayList<>();

        int threadCount = 4;
        ForkJoinPool myPool = new ForkJoinPool(threadCount);
        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
        //ParSort.cutoff = 100000;
        //for (int j = 1 ; j < 20 ; j++){
        long time;
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:10000    100times Time:10012ms
cutoff:20000    100times Time:9071ms
cutoff:30000    100times Time:8904ms
cutoff:40000    100times Time:8727ms
cutoff:50000    100times Time:8708ms
cutoff:60000    100times Time:8714ms
cutoff:70000    100times Time:8599ms
cutoff:80000    100times Time:8551ms
cutoff:90000    100times Time:8558ms
cutoff:100000   100times Time:8555ms
cutoff:110000   100times Time:8593ms
cutoff:120000   100times Time:8613ms
cutoff:130000   100times Time:8587ms
cutoff:140000   100times Time:8627ms
cutoff:150000   100times Time:8575ms

Process finished with exit code 0
```



```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[2500000];
        ArrayList<Long> timeList = new ArrayList<>();

        int threadCount = 4;
        ForkJoinPool myPool = new ForkJoinPool(threadCount);
        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
```

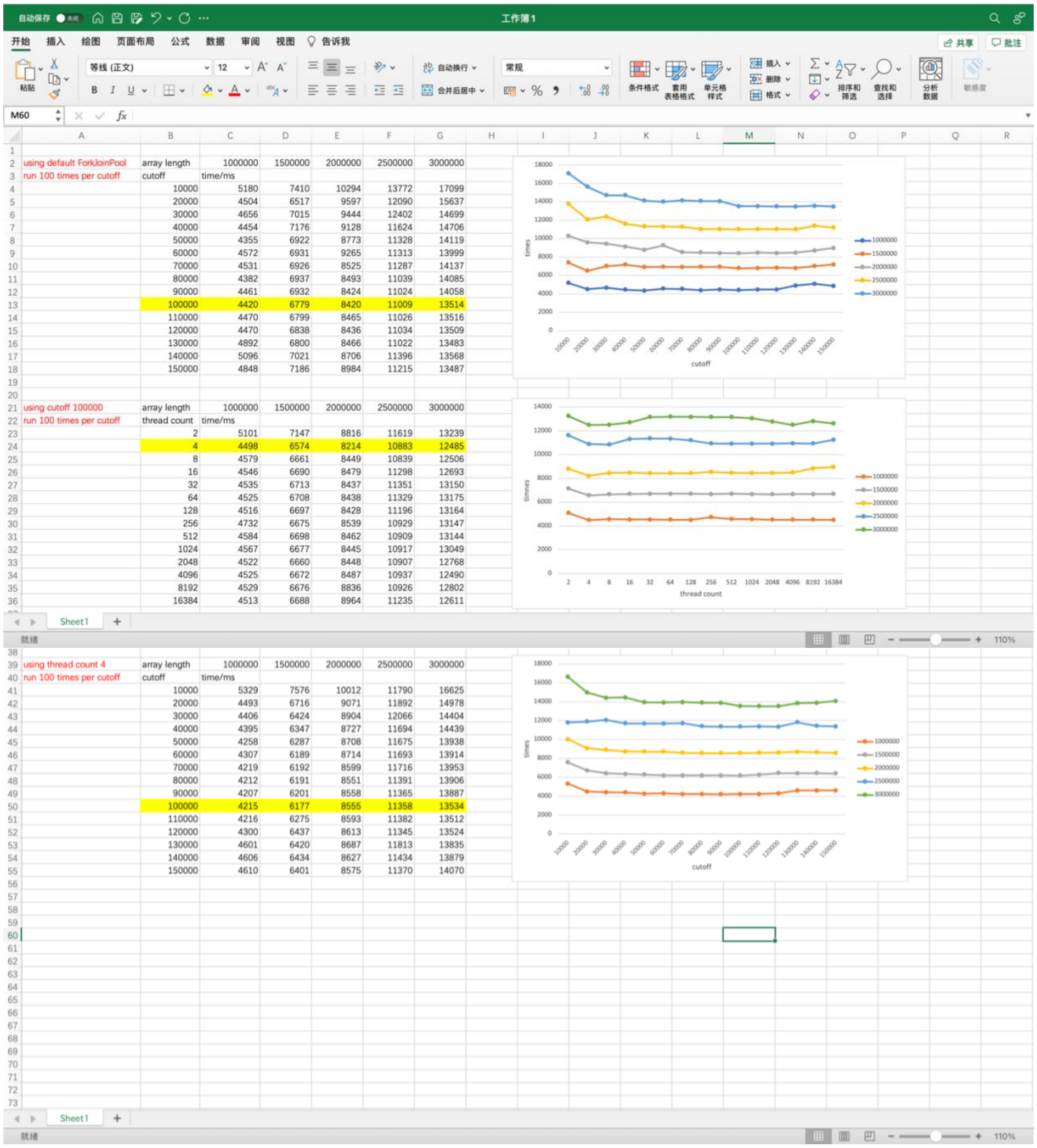
```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:10000    100times Time:11790ms
cutoff:20000    100times Time:11892ms
cutoff:30000    100times Time:12066ms
cutoff:40000    100times Time:11694ms
cutoff:50000    100times Time:11675ms
cutoff:60000    100times Time:11693ms
cutoff:70000    100times Time:11716ms
cutoff:80000    100times Time:11391ms
cutoff:90000    100times Time:11365ms
cutoff:100000   100times Time:11358ms
cutoff:110000   100times Time:11382ms
cutoff:120000   100times Time:11345ms
cutoff:130000   100times Time:11813ms
cutoff:140000   100times Time:11434ms
cutoff:150000   100times Time:11370ms

Process finished with exit code 0
```



```java
public class Main {

    public static void main(String[] args) {
        processArgs(args);
        System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolParallelism());
        Random random = new Random();
        int[] array = new int[3000000];
        ArrayList<Long> timeList = new ArrayList<>();

        int threadCount = 4;
        ForkJoinPool myPool = new ForkJoinPool(threadCount);
        for (int j = 1; j < 16; j++) {
            ParSort.cutoff = 10000 * j ;
```

```
"/Applications/IntelliJ IDEA CE.app/Contents/jbr/Contents/Home/bin/java" ...
Degree of parallelism: 7
cutoff:10000    100times Time:16625ms
cutoff:20000    100times Time:14978ms
cutoff:30000    100times Time:14404ms
cutoff:40000    100times Time:14439ms
cutoff:50000    100times Time:13938ms
cutoff:60000    100times Time:13914ms
cutoff:70000    100times Time:13953ms
cutoff:80000    100times Time:13906ms
cutoff:90000    100times Time:13887ms
cutoff:100000   100times Time:13534ms
cutoff:110000   100times Time:13512ms
cutoff:120000   100times Time:13524ms
cutoff:130000   100times Time:13835ms
cutoff:140000   100times Time:13879ms
cutoff:150000   100times Time:14070ms

Process finished with exit code 0
```

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



using default ForkJoinPool	array length	1000000	1500000	2000000	2500000	3000000
run 100 times per cutoff	cutoff	time/ms				
	10000	5180	7410	10294	13772	17099
	20000	4504	6517	9597	12090	15637
	30000	4656	7015	9444	12402	14699
	40000	4454	7176	9128	11624	14706
	50000	4355	6922	8773	11328	14119
	60000	4572	6931	9265	11313	13999
	70000	4531	6926	8525	11287	14137
	80000	4382	6937	8493	11039	14085
	90000	4461	6932	8424	11024	14058
	100000	4420	6779	8420	11009	13514
	110000	4470	6799	8465	11026	13516
	120000	4470	6838	8436	11034	13509
	130000	4892	6800	8466	11022	13483
	140000	5096	7021	8706	11396	13568
	150000	4848	7186	8984	11215	13487

using cutoff 100000	array length	1000000	1500000	2000000	2500000	3000000
run 100 times per cutoff	thread count	time/ms				
	2	5101	7147	8816	11619	13239
	4	4498	6574	8214	10883	12485
	8	4579	6661	8449	10839	12506
	16	4546	6690	8479	11298	12693
	32	4535	6713	8437	11351	13150
	64	4525	6708	8438	11329	13175
	128	4516	6697	8428	11196	13164
	256	4732	6675	8539	10929	13147
	512	4584	6698	8462	10909	13144
	1024	4567	6677	8445	10917	13049
	2048	4522	6660	8448	10907	12768
	4096	4525	6672	8487	10937	12490
	8192	4529	6676	8836	10926	12802
	16384	4513	6688	8964	11235	12611

using thread count 4	array length	1000000	1500000	2000000	2500000	3000000
run 100 times per cutoff	cutoff	time/ms				
	10000	5329	7576	10012	11790	16625
	20000	4493	6716	9071	11892	14978
	30000	4406	6424	8904	12066	14404
	40000	4395	6347	8727	11694	14439
	50000	4258	6287	8708	11675	13938
	60000	4307	6189	8714	11693	13914
	70000	4219	6192	8599	11716	13953
	80000	4212	6191	8551	11391	13906
	90000	4207	6201	8558	11365	13887
	100000	4215	6177	8555	11358	13534
	110000	4216	6275	8593	11382	13512
	120000	4300	6437	8613	11345	13524
	130000	4601	6420	8687	11813	13835
	140000	4606	6434	8627	11434	13879
	150000	4610	6401	8575	11370	14070