SANTHOSH KRISHNA MADDI

NUID : 002127881

INFO 6205 - ASSIGNMENT 5

Task:

- To implement a parallel sorting algorithm such that each partition of the array is sorted in parallel.
- Experiment and come up with a good value for the cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead of parallel sort.
- Decide on an ideal number (*t*) of separate threads and arrange for that number of partitions to be parallelized.

Conclusion:

After Conducting experiment for different array sizes considering large range of cut off values, the time taken for sorting the array is better when the cutoff value is close to the half of the array size.

Findings:

The experiment is conducted considering four different large values for array size and various cut of values ranging from 20000 to 100000 with few different degrees of parallelism. Below are the experiment values for each array size.

For each array size and cut off value the array is sorted 10 times and the time taken for ten time is displayed.

Degree of parallelism: 4 Array Size = 2000000

```
cutoff : 20000       10times Time:1783ms
cutoff : 60000       10times Time:855ms
cutoff : 100000      10times Time:838ms
cutoff : 140000      10times Time:871ms
cutoff : 180000      10times Time:1062ms
cutoff : 220000      10times Time:608ms
cutoff : 260000      10times Time:609ms
cutoff : 300000      10times Time:626ms
cutoff : 340000      10times Time:609ms
cutoff : 380000      10times Time:626ms
cutoff : 420000      10times Time:609ms
cutoff : 460000      10times Time:617ms
cutoff : 500000      10times Time:621ms
cutoff : 540000      10times Time:606ms
cutoff : 580000      10times Time:609ms
cutoff : 620000      10times Time:629ms
cutoff : 660000      10times Time:610ms
cutoff : 700000      10times Time:594ms
cutoff : 740000      10times Time:619ms
cutoff : 780000      10times Time:617ms
cutoff : 820000      10times Time:612ms
cutoff : 860000      10times Time:610ms
cutoff : 900000      10times Time:578ms
cutoff : 940000      10times Time:593ms
cutoff : 980000      10times Time:595m
```

Degree of parallelism: 4 Array Size = 2500000

```
cutoff：20000       10times Time:2213ms
cutoff：60000       10times Time:1236ms
cutoff：100000      10times Time:1262ms
cutoff：140000      10times Time:806ms
cutoff：180000      10times Time:880ms
cutoff：220000      10times Time:1049ms
cutoff：260000      10times Time:1035ms
cutoff：300000      10times Time:1070ms
cutoff：340000      10times Time:1067ms
cutoff：380000      10times Time:1098ms
cutoff：420000      10times Time:1075ms
cutoff：460000      10times Time:1129ms
cutoff：500000      10times Time:1090ms
cutoff：540000      10times Time:1176ms
cutoff：580000      10times Time:1191ms
cutoff：620000      10times Time:1147ms
cutoff：660000      10times Time:845ms
cutoff：700000      10times Time:781ms
cutoff：740000      10times Time:760ms
cutoff：780000      10times Time:766ms
cutoff：820000      10times Time:762ms
cutoff：860000      10times Time:800ms
cutoff：900000      10times Time:756ms
cutoff：940000      10times Time:750ms
cutoff：980000      10times Time:769ms
```

Degree of parallelism: 4   Array Size = 3000000

```
cutoff：20000        10times Time:2421ms
cutoff：60000        10times Time:1233ms
cutoff：100000       10times Time:1245ms
cutoff：140000       10times Time:1231ms
cutoff：180000       10times Time:1211ms
cutoff：220000       10times Time:1258ms
cutoff：260000       10times Time:1271ms
cutoff：300000       10times Time:1272ms
cutoff：340000       10times Time:1247ms
cutoff：380000       10times Time:1313ms
cutoff：420000       10times Time:1298ms
cutoff：460000       10times Time:1292ms
cutoff：500000       10times Time:1303ms
cutoff：540000       10times Time:1329ms
cutoff：580000       10times Time:995ms
cutoff：620000       10times Time:952ms
cutoff：660000       10times Time:938ms
cutoff：700000       10times Time:938ms
cutoff：740000       10times Time:950ms
cutoff：780000       10times Time:906ms
cutoff：820000       10times Time:906ms
cutoff：860000       10times Time:877ms
cutoff：900000       10times Time:891ms
cutoff：940000       10times Time:919ms
cutoff：980000       10times Time:887ms
```

Degree of parallelism: 4 Array Size = 3500000

cutoff：20000	10times Time:2916ms
cutoff：60000	10times Time:1862ms
cutoff：100000	10times Time:1641ms
cutoff：140000	10times Time:1675ms
cutoff：180000	10times Time:1647ms
cutoff：220000	10times Time:1525ms
cutoff：260000	10times Time:1073ms
cutoff：300000	10times Time:1051ms
cutoff：340000	10times Time:1027ms
cutoff：380000	10times Time:1185ms
cutoff：420000	10times Time:1432ms
cutoff：460000	10times Time:1497ms
cutoff：500000	10times Time:1468ms
cutoff：540000	10times Time:1529ms
cutoff：580000	10times Time:1470ms
cutoff：620000	10times Time:1500ms
cutoff：660000	10times Time:1453ms
cutoff：700000	10times Time:1492ms
cutoff：740000	10times Time:1493ms
cutoff：780000	10times Time:1666ms
cutoff：820000	10times Time:1478ms
cutoff：860000	10times Time:1499ms
cutoff：900000	10times Time:1438ms
cutoff：940000	10times Time:1202ms
cutoff：980000	10times Time:1047ms

Degree of parallelism: 2 Array Size = 2000000

```
cutoff：20000        10times Time:1534ms
cutoff：60000        10times Time:828ms
cutoff：100000       10times Time:847ms
cutoff：140000       10times Time:875ms
cutoff：180000       10times Time:907ms
cutoff：220000       10times Time:1057ms
cutoff：260000       10times Time:965ms
cutoff：300000       10times Time:1028ms
cutoff：340000       10times Time:985ms
cutoff：380000       10times Time:991ms
cutoff：420000       10times Time:970ms
cutoff：460000       10times Time:993ms
cutoff：500000       10times Time:1141ms
cutoff：540000       10times Time:791ms
cutoff：580000       10times Time:926ms
cutoff：620000       10times Time:853ms
cutoff：660000       10times Time:850ms
cutoff：700000       10times Time:790ms
cutoff：740000       10times Time:767ms
cutoff：780000       10times Time:751ms
cutoff：820000       10times Time:767ms
cutoff：860000       10times Time:767ms
cutoff：900000       10times Time:749ms
cutoff：940000       10times Time:766ms
cutoff：980000       10times Time:959ms
```

Degree of parallelism: 2 Array Size = 2500000

```
cutoff：20000        10times Time:1995ms
cutoff：60000        10times Time:1099ms
cutoff：100000       10times Time:1056ms
cutoff：140000       10times Time:1067ms
cutoff：180000       10times Time:1151ms
cutoff：220000       10times Time:1153ms
cutoff：260000       10times Time:1132ms
cutoff：300000       10times Time:1340ms
cutoff：340000       10times Time:1654ms
cutoff：380000       10times Time:1503ms
cutoff：420000       10times Time:1025ms
cutoff：460000       10times Time:895ms
cutoff：500000       10times Time:909ms
cutoff：540000       10times Time:909ms
cutoff：580000       10times Time:898ms
cutoff：620000       10times Time:906ms
cutoff：660000       10times Time:955ms
cutoff：700000       10times Time:978ms
cutoff：740000       10times Time:978ms
cutoff：780000       10times Time:1030ms
cutoff：820000       10times Time:1037ms
cutoff：860000       10times Time:970ms
cutoff：900000       10times Time:957ms
cutoff：940000       10times Time:979ms
cutoff：980000       10times Time:987ms
```

Degree of parallelism: 2 Array Size = 3000000

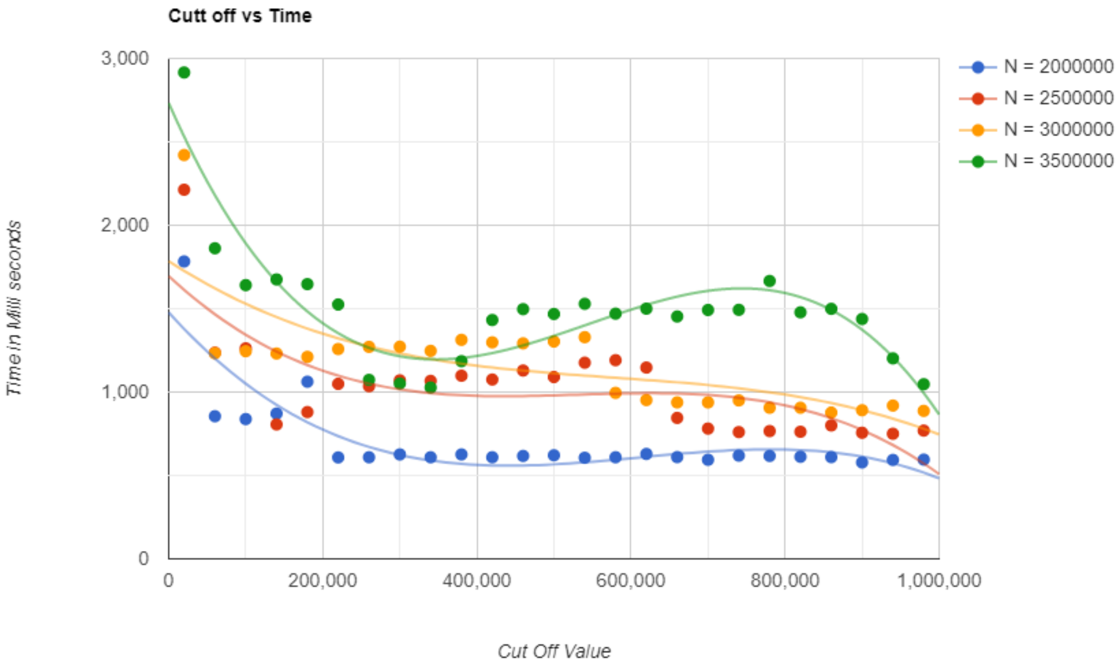
cutoff：20000	10times Time:2307ms
cutoff：60000	10times Time:1251ms
cutoff：100000	10times Time:1282ms
cutoff：140000	10times Time:1243ms
cutoff：180000	10times Time:1230ms
cutoff：220000	10times Time:1472ms
cutoff：260000	10times Time:1572ms
cutoff：300000	10times Time:1625ms
cutoff：340000	10times Time:1563ms
cutoff：380000	10times Time:1491ms
cutoff：420000	10times Time:1454ms
cutoff：460000	10times Time:1452ms
cutoff：500000	10times Time:1450ms
cutoff：540000	10times Time:1444ms
cutoff：580000	10times Time:1478ms
cutoff：620000	10times Time:1558ms
cutoff：660000	10times Time:1473ms
cutoff：700000	10times Time:1471ms
cutoff：740000	10times Time:1474ms
cutoff：780000	10times Time:1533ms
cutoff：820000	10times Time:1645ms
cutoff：860000	10times Time:1658ms
cutoff：900000	10times Time:1560ms
cutoff：940000	10times Time:1617ms
cutoff：980000	10times Time:1580ms

Degree of parallelism: 2 Array Size = 3500000

```
cutoff：20000        10times Time:2815ms
cutoff：60000        10times Time:1572ms
cutoff：100000       10times Time:1617ms
cutoff：140000       10times Time:1799ms
cutoff：180000       10times Time:1794ms
cutoff：220000       10times Time:1795ms
cutoff：260000       10times Time:2219ms
cutoff：300000       10times Time:2070ms
cutoff：340000       10times Time:2117ms
cutoff：380000       10times Time:1916ms
cutoff：420000       10times Time:2078ms
cutoff：460000       10times Time:1996ms
cutoff：500000       10times Time:2090ms
cutoff：540000       10times Time:1887ms
cutoff：580000       10times Time:1785ms
cutoff：620000       10times Time:1929ms
cutoff：660000       10times Time:1930ms
cutoff：700000       10times Time:2048ms
cutoff：740000       10times Time:1879ms
cutoff：780000       10times Time:1724ms
cutoff：820000       10times Time:1845ms
cutoff：860000       10times Time:1909ms
cutoff：900000       10times Time:1845ms
cutoff：940000       10times Time:1894ms
cutoff：980000       10times Time:1920ms
```

Graph:

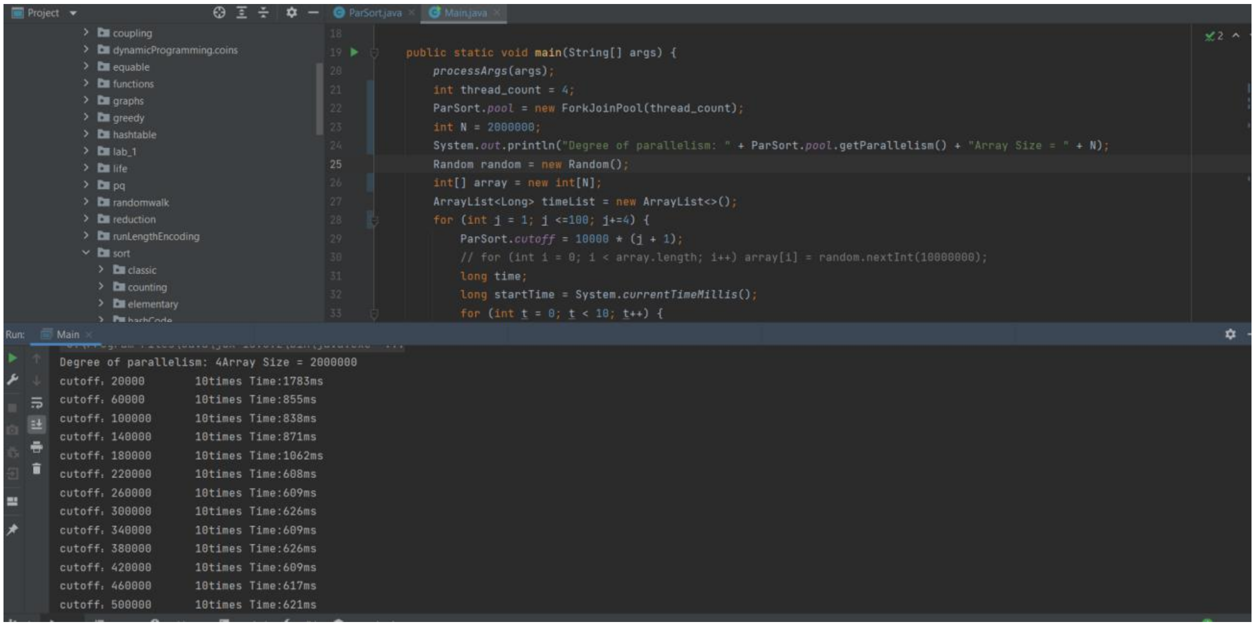
Below is the graph plotted for degree of parallelism value 4 for different array sizes(N) with wide range of cut of values.



From the above graph the time taken to sort array 10 times is better at 100000 value which is closer to the half of the array size.

Output Screenshots:

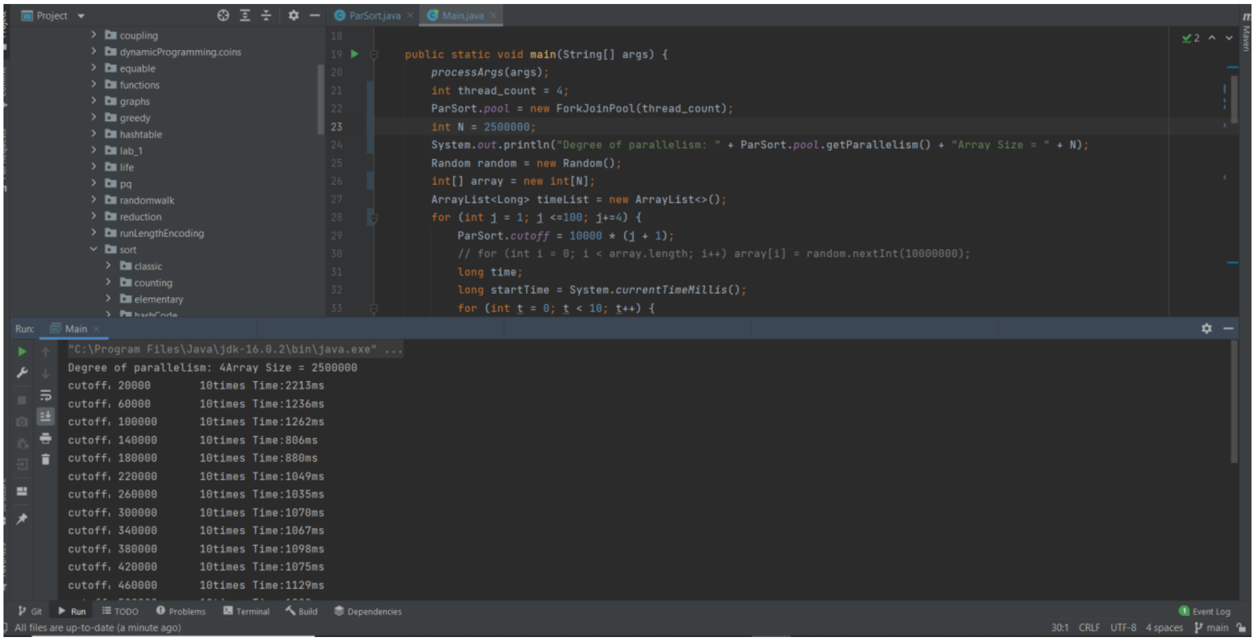
Degree of parallelism 4 , Array Size = 2000000



```java
    public static void main(String[] args) {
        processArgs(args);
        int thread_count = 4;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 2000000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
        Random random = new Random();
        int[] array = new int[N];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j <=100; j+=4) {
            ParSort.cutoff = 10000 * (j + 1);
            // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
            long time;
            long startTime = System.currentTimeMillis();
            for (int t = 0; t < 10; t++) {
```

```
Degree of parallelism: 4Array Size = 2000000
cutoff：20000       10times Time:1783ms
cutoff：60000       10times Time:855ms
cutoff：100000      10times Time:838ms
cutoff：140000      10times Time:871ms
cutoff：180000      10times Time:1062ms
cutoff：220000      10times Time:608ms
cutoff：260000      10times Time:609ms
cutoff：300000      10times Time:626ms
cutoff：340000      10times Time:609ms
cutoff：380000      10times Time:626ms
cutoff：420000      10times Time:609ms
cutoff：460000      10times Time:617ms
cutoff：500000      10times Time:621ms
```

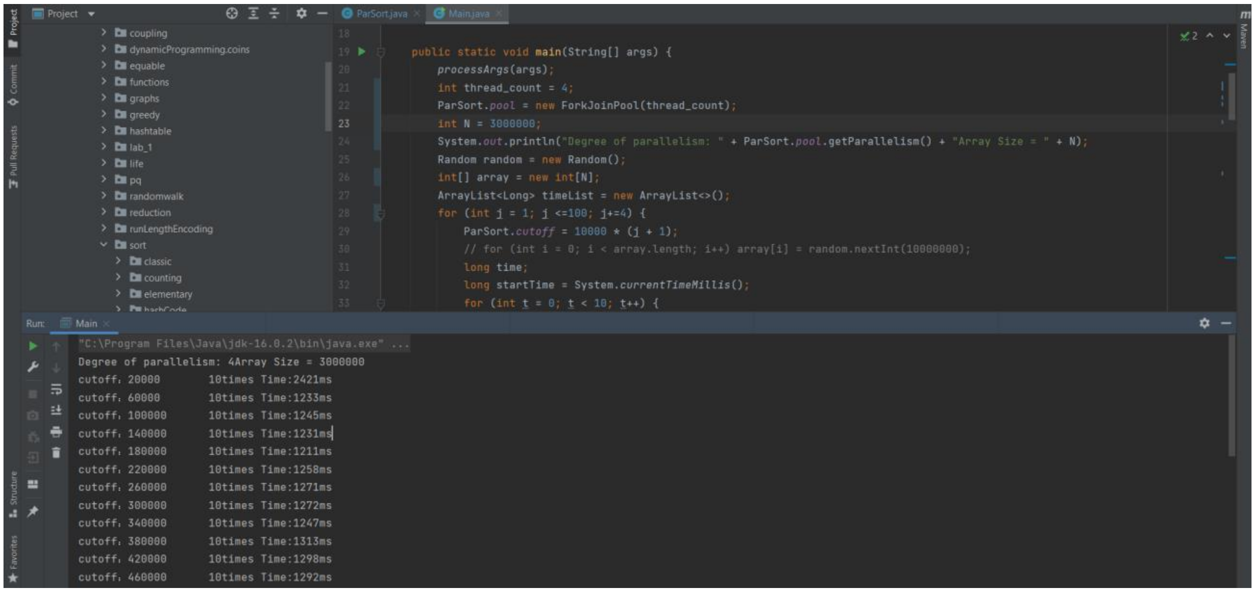
Degree of parallelism 4 , Array Size = 2500000



```java
    public static void main(String[] args) {
        processArgs(args);
        int thread_count = 4;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 2500000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
        Random random = new Random();
        int[] array = new int[N];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j <=100; j+=4) {
            ParSort.cutoff = 10000 * (j + 1);
            // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
            long time;
            long startTime = System.currentTimeMillis();
            for (int t = 0; t < 10; t++) {
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Degree of parallelism: 4Array Size = 2500000
cutoff：20000       10times Time:2213ms
cutoff：60000       10times Time:1236ms
cutoff：100000      10times Time:1262ms
cutoff：140000      10times Time:806ms
cutoff：180000      10times Time:880ms
cutoff：220000      10times Time:1049ms
cutoff：260000      10times Time:1035ms
cutoff：300000      10times Time:1070ms
cutoff：340000      10times Time:1067ms
cutoff：380000      10times Time:1098ms
cutoff：420000      10times Time:1075ms
cutoff：460000      10times Time:1129ms
```

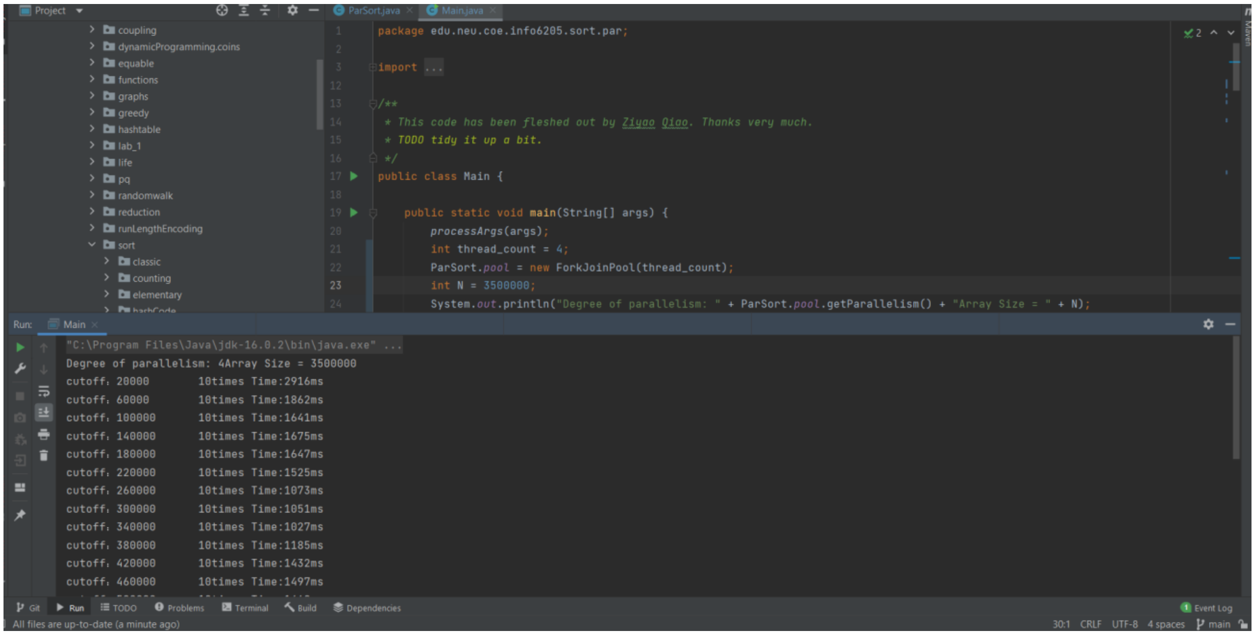
Degree of parallelism 4 , Array Size = 3000000



```java
    public static void main(String[] args) {
        processArgs(args);
        int thread_count = 4;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 3000000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
        Random random = new Random();
        int[] array = new int[N];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j <=100; j+=4) {
            ParSort.cutoff = 10000 * (j + 1);
            // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
            long time;
            long startTime = System.currentTimeMillis();
            for (int t = 0; t < 10; t++) {
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Degree of parallelism: 4Array Size = 3000000
cutoff：20000       10times Time:2421ms
cutoff：60000       10times Time:1233ms
cutoff：100000      10times Time:1245ms
cutoff：140000      10times Time:1231ms
cutoff：180000      10times Time:1211ms
cutoff：220000      10times Time:1258ms
cutoff：260000      10times Time:1271ms
cutoff：300000      10times Time:1272ms
cutoff：340000      10times Time:1247ms
cutoff：380000      10times Time:1313ms
cutoff：420000      10times Time:1298ms
cutoff：460000      10times Time:1292ms
```

Degree of parallelism 4 , Array Size = 3500000

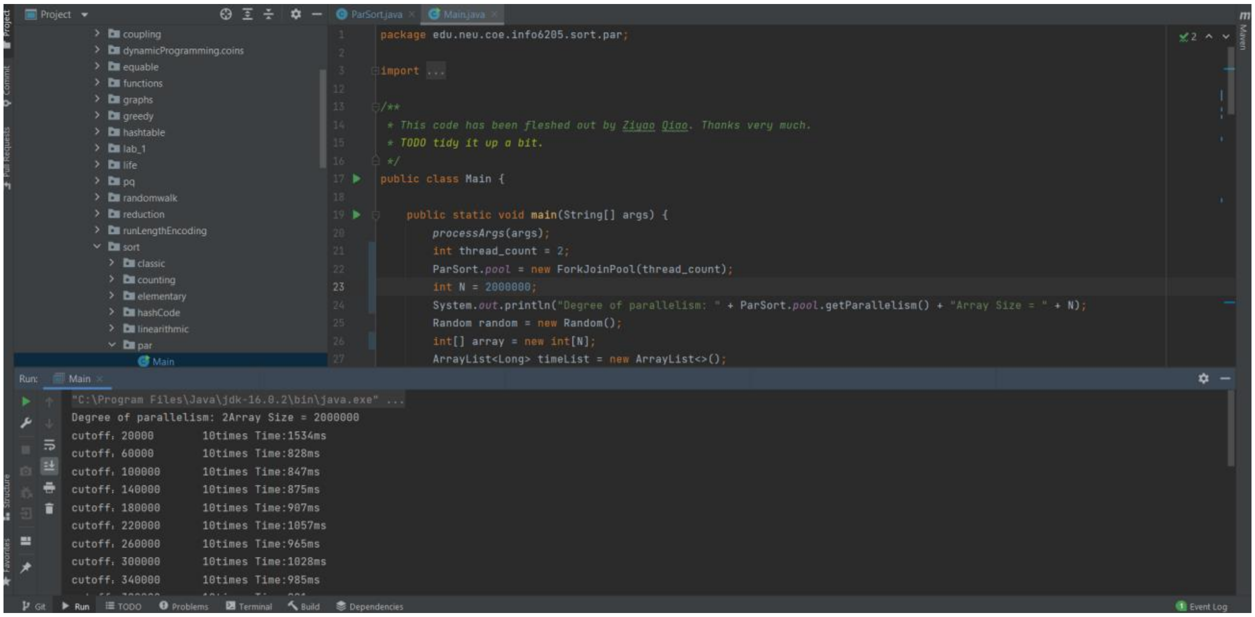


```java
package edu.neu.coe.info6205.sort.par;

import ...

/**
 * 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);
        int thread_count = 4;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 3500000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Degree of parallelism: 4Array Size = 3500000
cutoff：20000       10times Time:2916ms
cutoff：60000       10times Time:1862ms
cutoff：100000      10times Time:1641ms
cutoff：140000      10times Time:1675ms
cutoff：180000      10times Time:1647ms
cutoff：220000      10times Time:1525ms
cutoff：260000      10times Time:1073ms
cutoff：300000      10times Time:1051ms
cutoff：340000      10times Time:1027ms
cutoff：380000      10times Time:1185ms
cutoff：420000      10times Time:1432ms
cutoff：460000      10times Time:1497ms
```

All files are up-to-date (a minute ago)

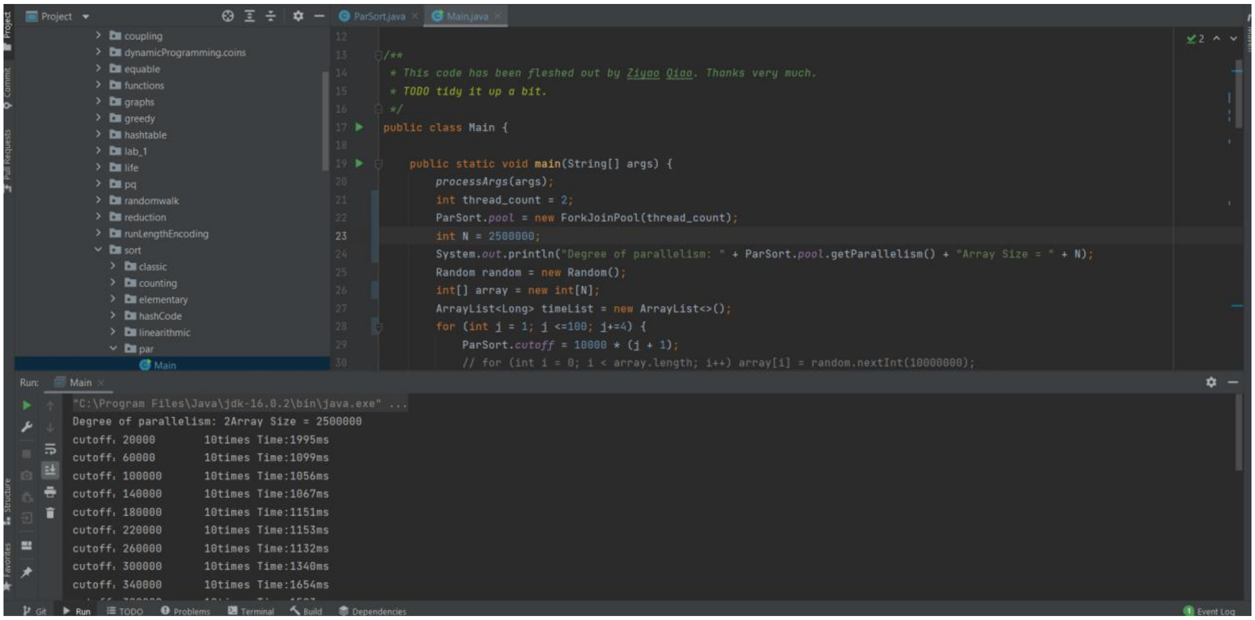
Degree of parallelism 2 , Array Size = 2000000



```java
package edu.neu.coe.info6205.sort.par;

import ...

/**
 * 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);
        int thread_count = 2;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 2000000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
        Random random = new Random();
        int[] array = new int[N];
        ArrayList<Long> timeList = new ArrayList<>();
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Degree of parallelism: 2Array Size = 2000000
cutoff：20000       10times Time:1534ms
cutoff：60000       10times Time:828ms
cutoff：100000      10times Time:847ms
cutoff：140000      10times Time:875ms
cutoff：180000      10times Time:907ms
cutoff：220000      10times Time:1057ms
cutoff：260000      10times Time:965ms
cutoff：300000      10times Time:1028ms
cutoff：340000      10times Time:985ms
```

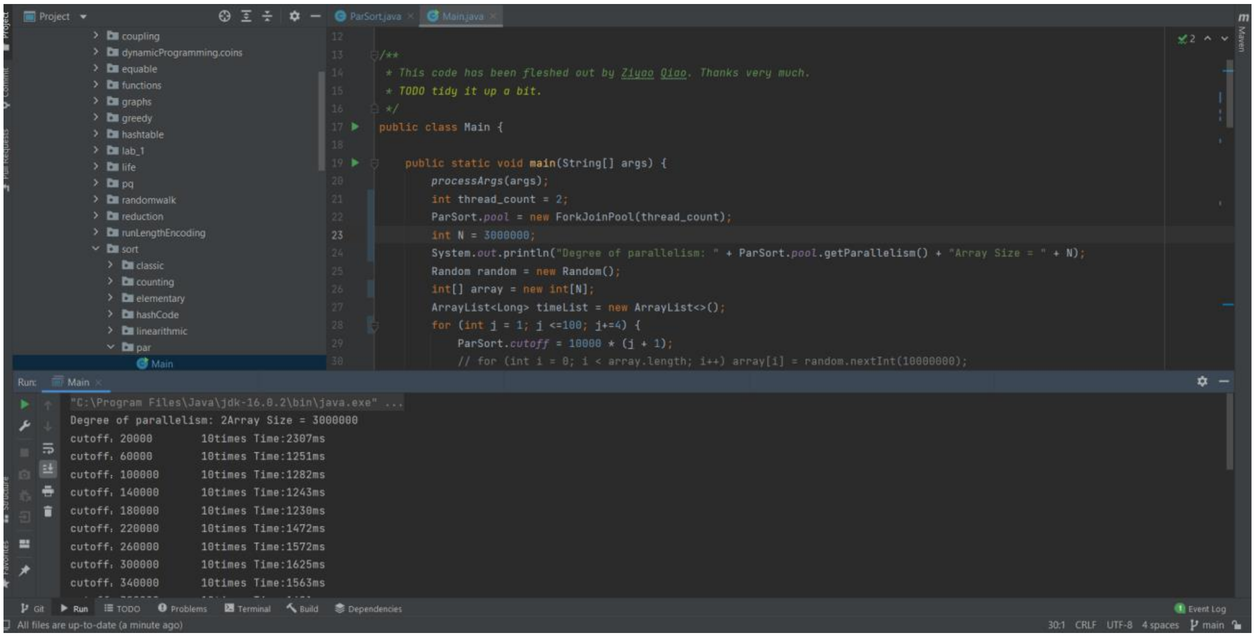
Degree of parallelism 2 , Array Size = 2500000



```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);
        int thread_count = 2;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 2500000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
        Random random = new Random();
        int[] array = new int[N];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j <=100; j+=4) {
            ParSort.cutoff = 10000 * (j + 1);
            // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Degree of parallelism: 2Array Size = 2500000
cutoff：20000       10times Time:1995ms
cutoff：60000       10times Time:1099ms
cutoff：100000      10times Time:1056ms
cutoff：140000      10times Time:1067ms
cutoff：180000      10times Time:1151ms
cutoff：220000      10times Time:1153ms
cutoff：260000      10times Time:1132ms
cutoff：300000      10times Time:1340ms
cutoff：340000      10times Time:1654ms
```

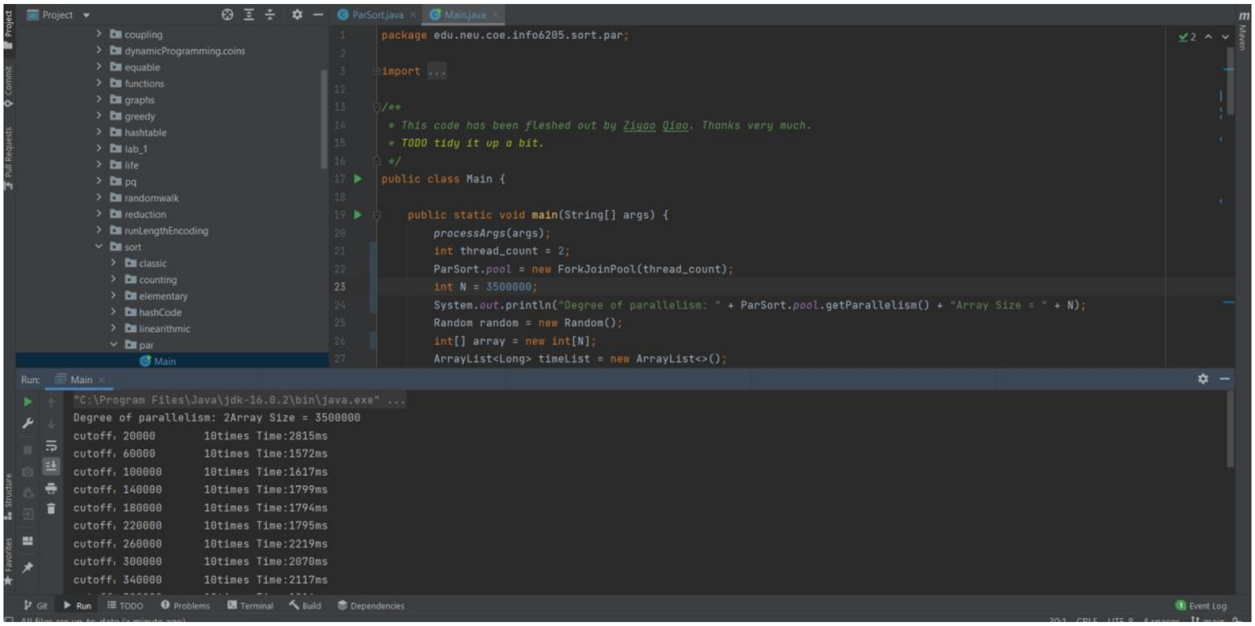
Degree of parallelism 2 , Array Size = 3000000



```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);
        int thread_count = 2;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 3000000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
        Random random = new Random();
        int[] array = new int[N];
        ArrayList<Long> timeList = new ArrayList<>();
        for (int j = 1; j <=100; j+=4) {
            ParSort.cutoff = 10000 * (j + 1);
            // for (int i = 0; i < array.length; i++) array[i] = random.nextInt(10000000);
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Degree of parallelism: 2Array Size = 3000000
cutoff：20000       10times Time:2307ms
cutoff：60000       10times Time:1251ms
cutoff：100000      10times Time:1282ms
cutoff：140000      10times Time:1243ms
cutoff：180000      10times Time:1230ms
cutoff：220000      10times Time:1472ms
cutoff：260000      10times Time:1572ms
cutoff：300000      10times Time:1625ms
cutoff：340000      10times Time:1563ms
```

Degree of parallelism 2 , Array Size = 3500000



```java
package edu.neu.coe.info6205.sort.par;

import ...

/**
 * 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);
        int thread_count = 2;
        ParSort.pool = new ForkJoinPool(thread_count);
        int N = 3500000;
        System.out.println("Degree of parallelism: " + ParSort.pool.getParallelism() + "Array Size = " + N);
        Random random = new Random();
        int[] array = new int[N];
        ArrayList<Long> timeList = new ArrayList<>();
```

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" ...
Degree of parallelism: 2Array Size = 3500000
cutoff：20000       10times Time:2815ms
cutoff：60000       10times Time:1572ms
cutoff：100000      10times Time:1617ms
cutoff：140000      10times Time:1799ms
cutoff：180000      10times Time:1794ms
cutoff：220000      10times Time:1795ms
cutoff：260000      10times Time:2219ms
cutoff：300000      10times Time:2070ms
cutoff：340000      10times Time:2117ms
```