Program Structures and Algorithms Spring 2023
Section – 01
Name – Naman Diwan
NUID – 002724115

Assignment 5

Please see the presentation on Assignment on Parallel Sorting under the Exams. etc. module. **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.**

There is a Main class and the ParSort class in the sort.par package of the INFO6205 repository. The Main class can be used as is but the ParSort class needs to be implemented where you see "TODO..." [it turns out that these TODOs are already implemented].
Unless you have a good reason not to, you should just go along with the Java8-style future implementations provided for you in the class repository.
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

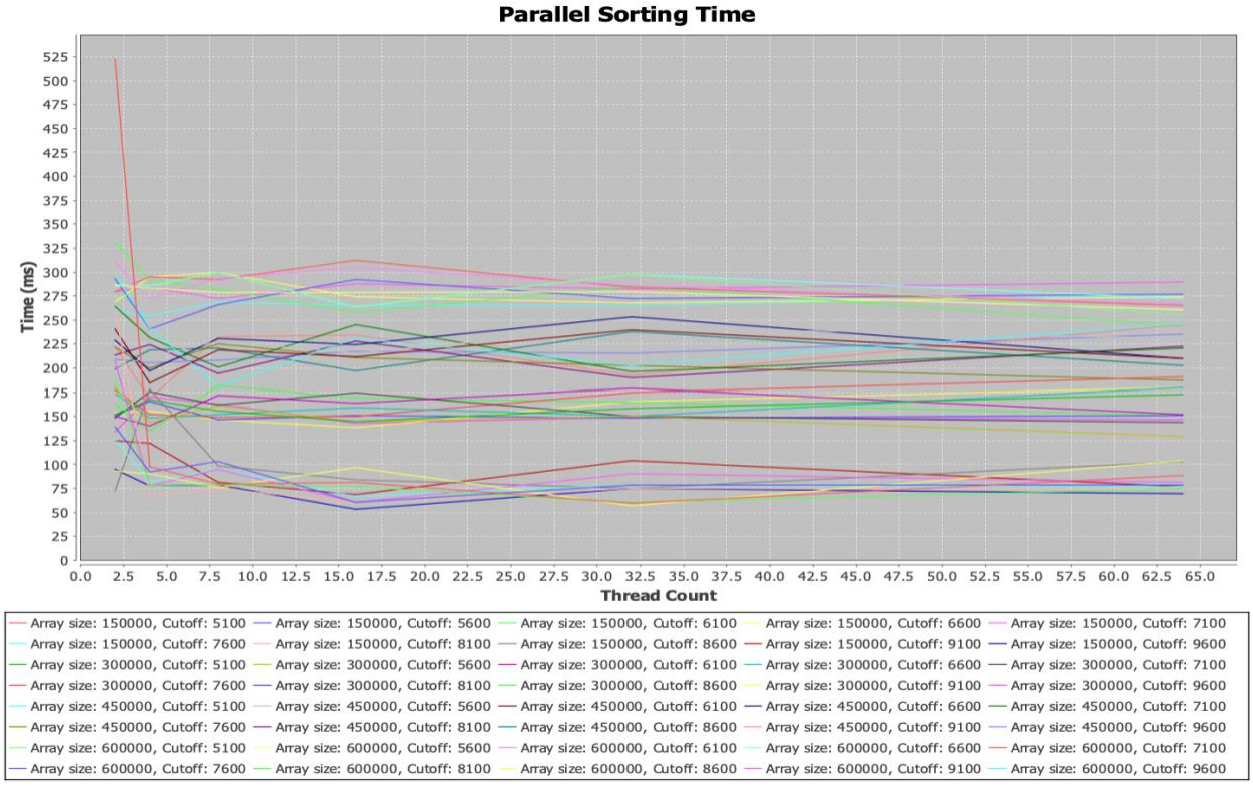
parallelizing sort. Your experiments should involve sorting arrays of sufficient size for the parallel sort to make a difference. You should run with many different array sizes (they must be sufficiently large to make parallel sorting worthwhile, obviously) and different cut-off schemes.

Code already uploaded to - https://github.com/namandiwan10/INFO-6205-Assignment-1

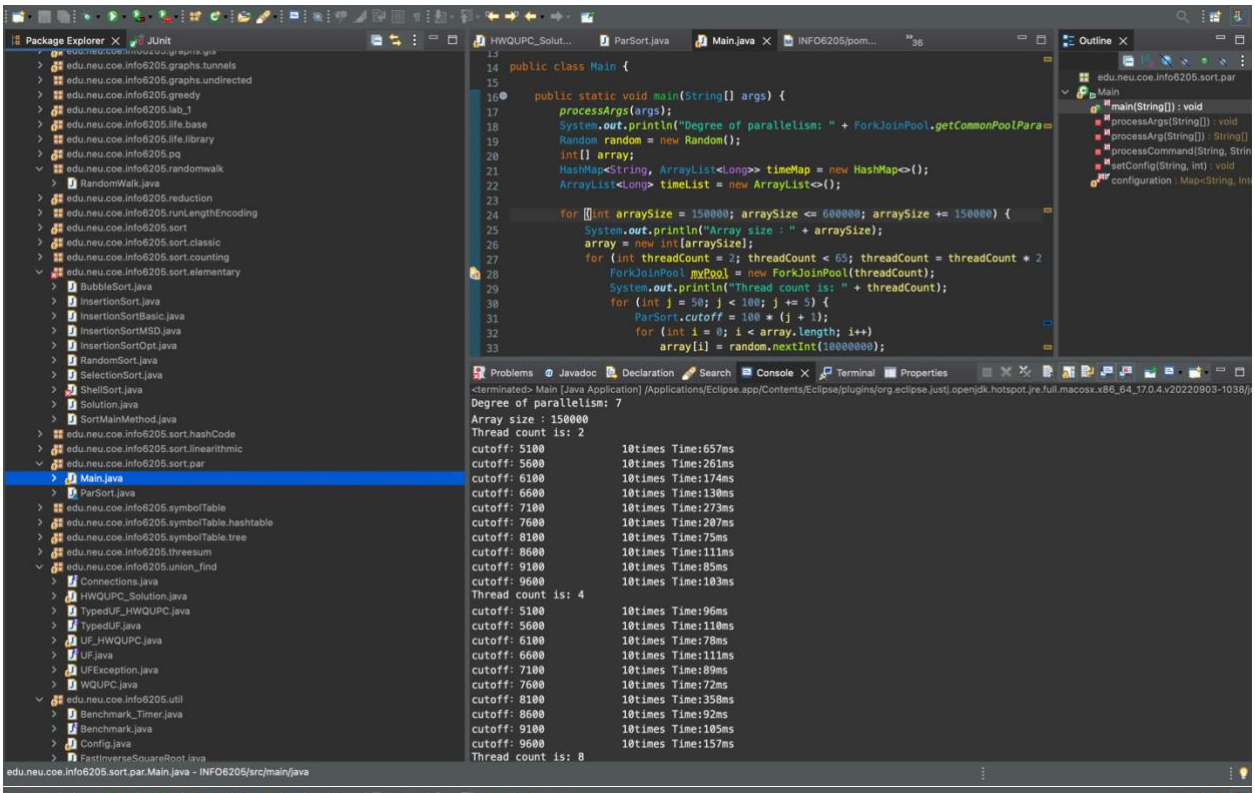
Observations

1. **As we increase the cut-off value, the time required to sort the array decreases, we can see in below graph .**
2. **As we increase the number of threads, the time required to sort the array also decreases.**
3. **As we increase the array size, the time required to sort the array also increases.**

Graph

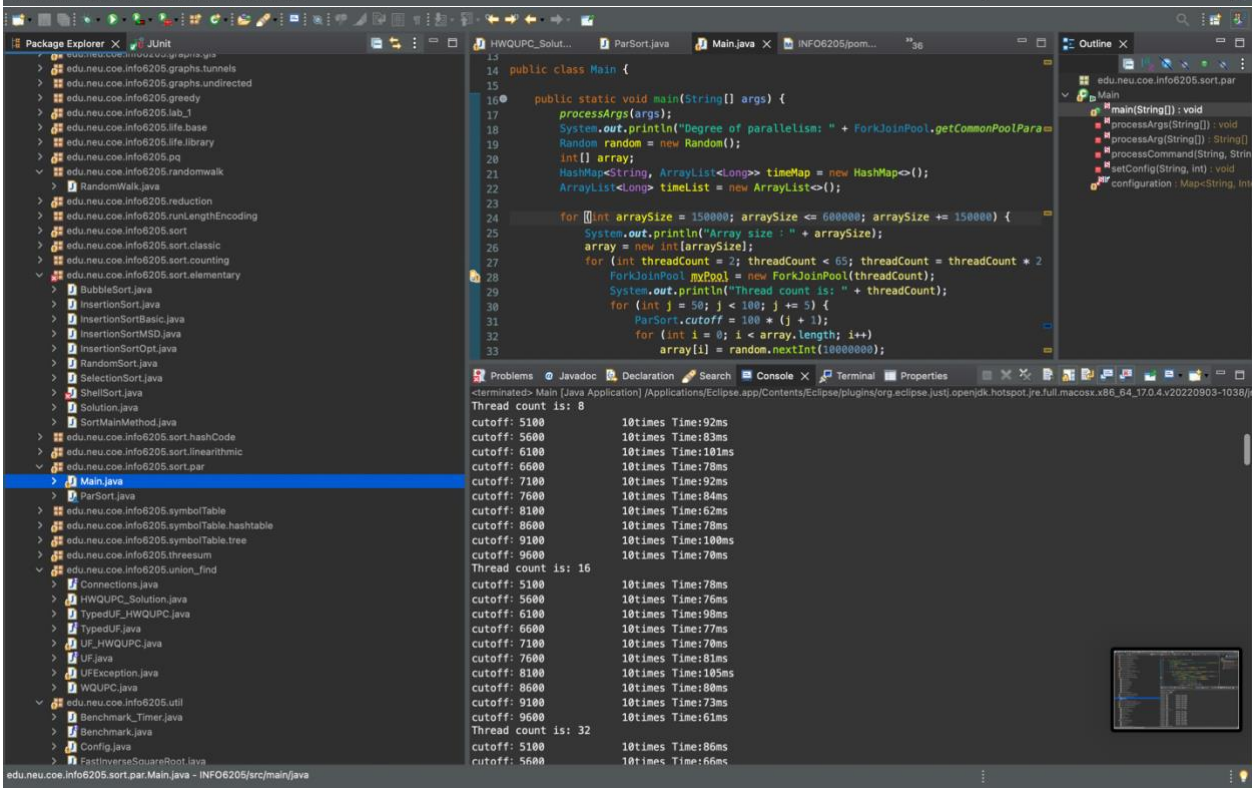


Readings



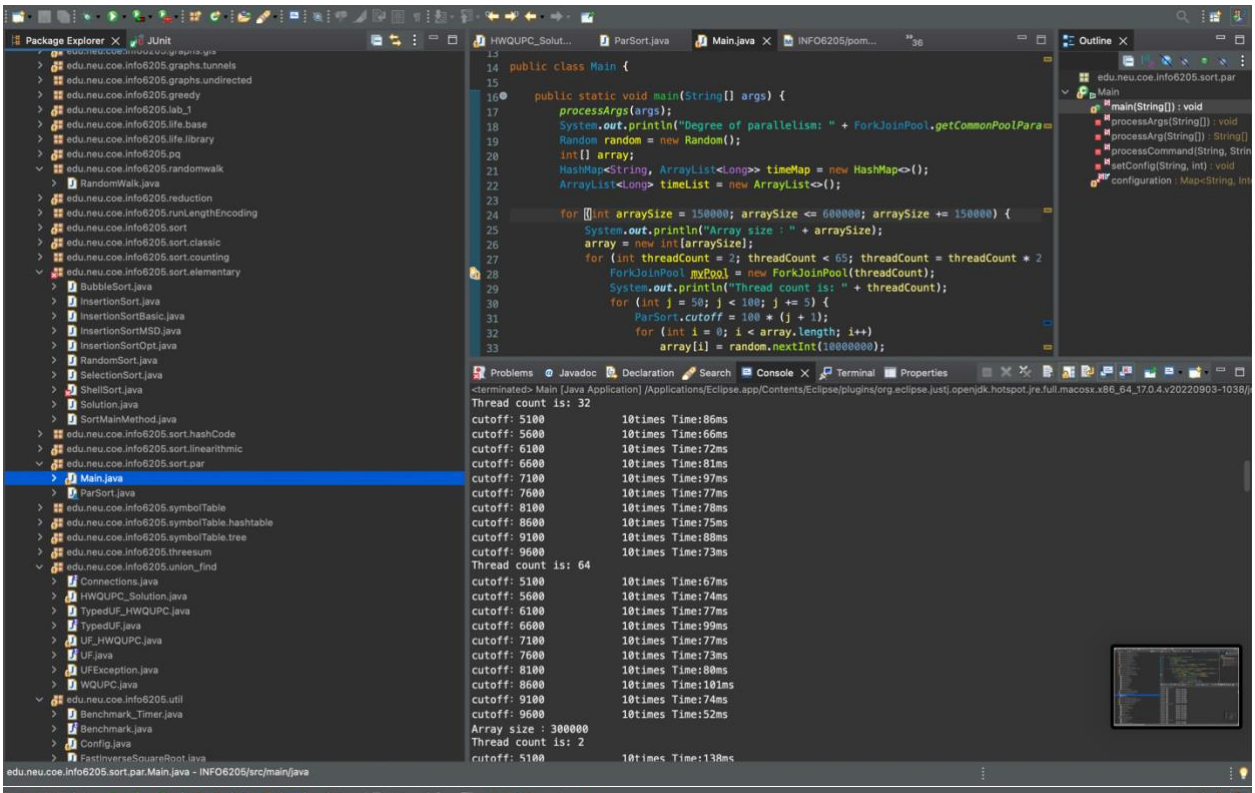

```
public class Main {

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

        for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
            System.out.println("Array size : " + arraySize);
            array = new int[arraySize];
            for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
                ForkJoinPool myPool = new ForkJoinPool(threadCount);
                System.out.println("Thread count is: " + threadCount);
                for (int j = 50; j < 100; j += 5) {
                    ParSort.cutoff = 100 * (j + 1);
                    for (int i = 0; i < array.length; i++)
                        array[i] = random.nextInt(10000000);
```

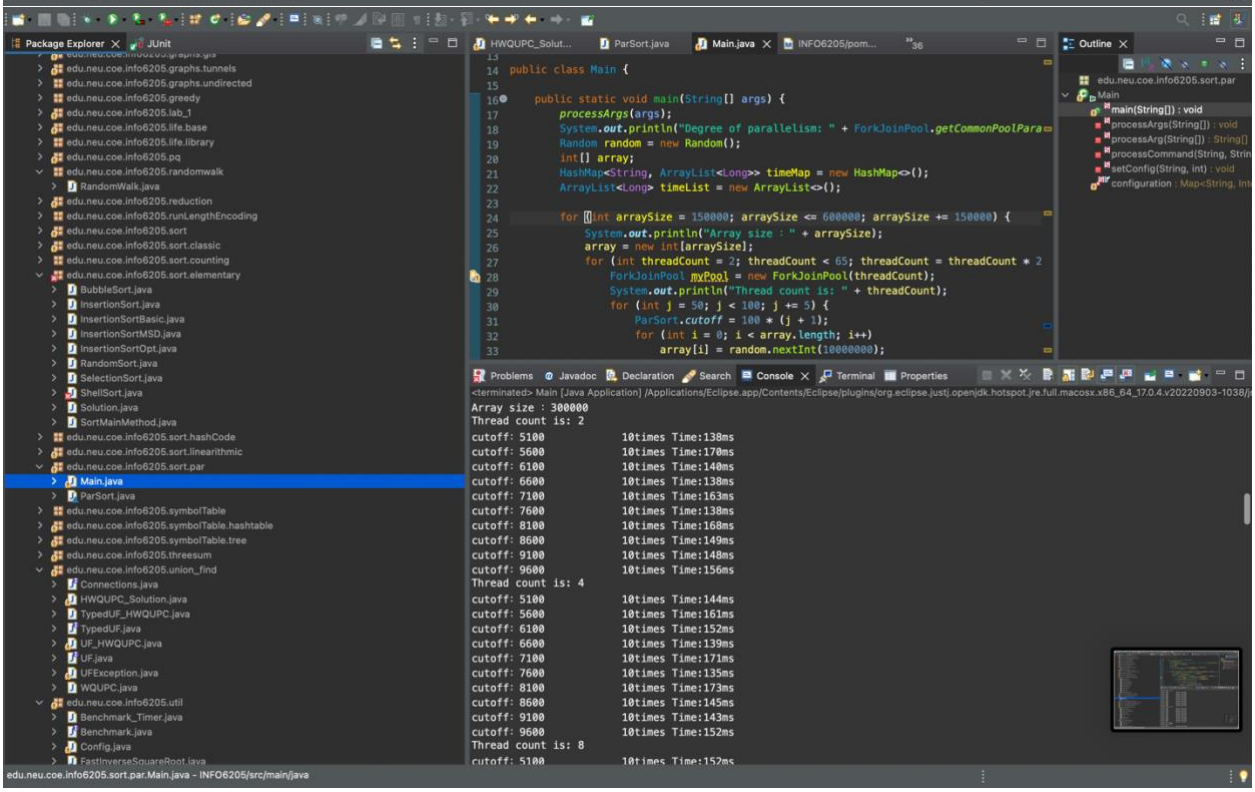
```
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Degree of parallelism: 7
Array size : 150000
Thread count is: 2
cutoff: 5100            10times Time:657ms
cutoff: 5600            10times Time:261ms
cutoff: 6100            10times Time:174ms
cutoff: 6600            10times Time:130ms
cutoff: 7100            10times Time:273ms
cutoff: 7600            10times Time:207ms
cutoff: 8100            10times Time:75ms
cutoff: 8600            10times Time:111ms
cutoff: 9100            10times Time:85ms
cutoff: 9600            10times Time:103ms
Thread count is: 4
cutoff: 5100            10times Time:96ms
cutoff: 5600            10times Time:110ms
cutoff: 6100            10times Time:78ms
cutoff: 6600            10times Time:111ms
cutoff: 7100            10times Time:89ms
cutoff: 7600            10times Time:72ms
cutoff: 8100            10times Time:358ms
cutoff: 8600            10times Time:92ms
cutoff: 9100            10times Time:105ms
cutoff: 9600            10times Time:157ms
Thread count is: 8
```

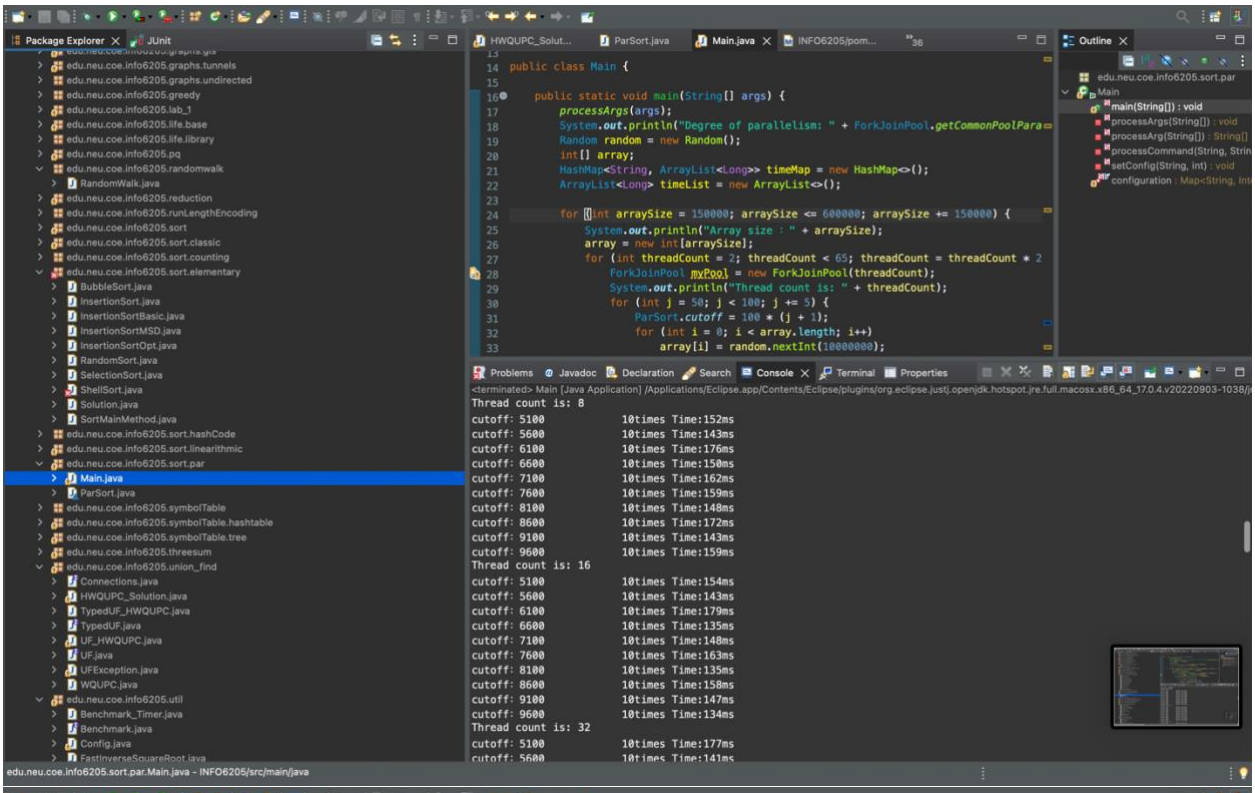



```
public class Main {

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

        for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
            System.out.println("Array size : " + arraySize);
            array = new int[arraySize];
            for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
                ForkJoinPool myPool = new ForkJoinPool(threadCount);
                System.out.println("Thread count is: " + threadCount);
                for (int j = 50; j < 100; j += 5) {
                    ParSort.cutoff = 100 * (j + 1);
                    for (int i = 0; i < array.length; i++)
                        array[i] = random.nextInt(10000000);
```

```
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Thread count is: 8
cutoff: 5100            10times Time:92ms
cutoff: 5600            10times Time:83ms
cutoff: 6100            10times Time:101ms
cutoff: 6600            10times Time:78ms
cutoff: 7100            10times Time:92ms
cutoff: 7600            10times Time:84ms
cutoff: 8100            10times Time:62ms
cutoff: 8600            10times Time:78ms
cutoff: 9100            10times Time:100ms
cutoff: 9600            10times Time:70ms
Thread count is: 16
cutoff: 5100            10times Time:78ms
cutoff: 5600            10times Time:76ms
cutoff: 6100            10times Time:90ms
cutoff: 6600            10times Time:77ms
cutoff: 7100            10times Time:70ms
cutoff: 7600            10times Time:81ms
cutoff: 8100            10times Time:105ms
cutoff: 8600            10times Time:80ms
cutoff: 9100            10times Time:73ms
cutoff: 9600            10times Time:61ms
Thread count is: 32
cutoff: 5100            10times Time:86ms
cutoff: 5600            10times Time:66ms
```

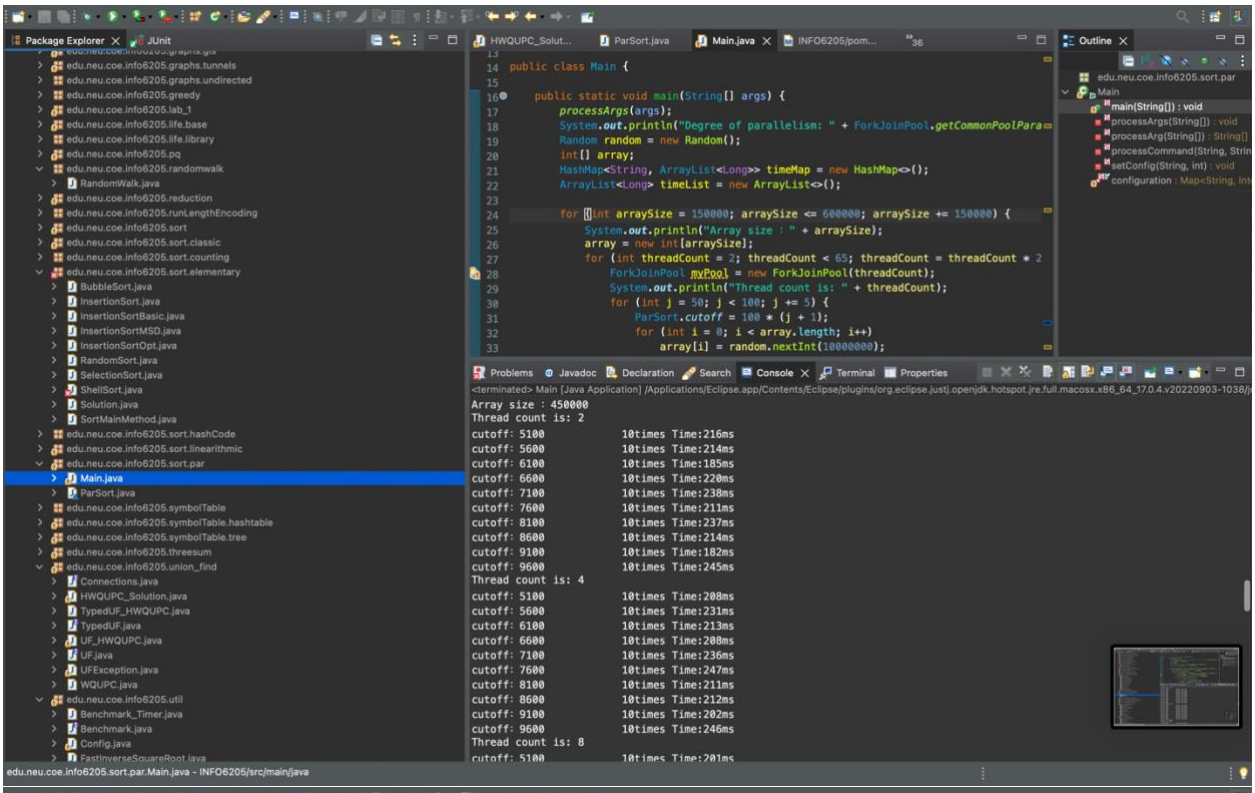

Package Explorer
JUnit
edu.neu.coe.info6205.graphs.tunnels
edu.neu.coe.info6205.graphs.undirected
edu.neu.coe.info6205.greedy
edu.neu.coe.info6205.lab_1
edu.neu.coe.info6205.life.base
edu.neu.coe.info6205.life.library
edu.neu.coe.info6205.pq
edu.neu.coe.info6205.randomwalk
RandomWalk.java
edu.neu.coe.info6205.reduction
edu.neu.coe.info6205.runLengthEncoding
edu.neu.coe.info6205.sort
edu.neu.coe.info6205.sort.classic
edu.neu.coe.info6205.sort.counting
edu.neu.coe.info6205.sort.elementary
BubbleSort.java
InsertionSort.java
InsertionSortBasic.java
InsertionSortMSD.java
InsertionSortOpt.java
RandomSort.java
SelectionSort.java
ShellSort.java
Solution.java
SortMainMethod.java
edu.neu.coe.info6205.sort.hashCode
edu.neu.coe.info6205.sort.linearithmic
edu.neu.coe.info6205.sort.par
Main.java
ParSort.java
edu.neu.coe.info6205.symbolTable
edu.neu.coe.info6205.symbolTable.hashtable
edu.neu.coe.info6205.symbolTable.tree
edu.neu.coe.info6205.threesum
edu.neu.coe.info6205.union_find
Connections.java
HWQUPC_Solution.java
TypedUF_HWQUPC.java
TypedUF.java
UF_HWQUPC.java
UF.java
UFException.java
WQUPC.java
edu.neu.coe.info6205.util
Benchmark_Timer.java
Benchmark.java
Config.java
FastInverseSquareRoot.java
HWQUPC_Solut...
ParSort.java
Main.java
INFO6205/pom...
36
public class Main {
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolPara
Random random = new Random();
int[] array;
HashMap<String, ArrayList<Long>> timeMap = new HashMap<>();
ArrayList<Long> timeList = new ArrayList<>();
for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
System.out.println("Array size : " + arraySize);
array = new int[arraySize];
for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
ForkJoinPool myPool = new ForkJoinPool(threadCount);
System.out.println("Thread count is: " + threadCount);
for (int j = 50; j < 100; j += 5) {
ParSort.cutoff = 100 * (j + 1);
for (int i = 0; i < array.length; i++)
array[i] = random.nextInt(10000000);
Outline
edu.neu.coe.info6205.sort.par
Main
main(String[]) : void
processArgs(String[]) : void
processArg(String[]) : String[]
processCommand(String, Strin
setConfig(String, int) : void
configuration : Map<String, Int
Problems
Javadoc
Declaration
Search
Console
Terminal
Properties
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Thread count is: 32
cutoff: 5100 10times Time:86ms
cutoff: 5600 10times Time:66ms
cutoff: 6100 10times Time:72ms
cutoff: 6600 10times Time:81ms
cutoff: 7100 10times Time:97ms
cutoff: 7600 10times Time:77ms
cutoff: 8100 10times Time:78ms
cutoff: 8600 10times Time:75ms
cutoff: 9100 10times Time:88ms
cutoff: 9600 10times Time:73ms
Thread count is: 64
cutoff: 5100 10times Time:67ms
cutoff: 5600 10times Time:74ms
cutoff: 6100 10times Time:77ms
cutoff: 6600 10times Time:99ms
cutoff: 7100 10times Time:77ms
cutoff: 7600 10times Time:73ms
cutoff: 8100 10times Time:80ms
cutoff: 8600 10times Time:101ms
cutoff: 9100 10times Time:74ms
cutoff: 9600 10times Time:52ms
Array size : 300000
Thread count is: 2
cutoff: 5100 10times Time:138ms
edu.neu.coe.info6205.sort.par.Main.java - INFO6205/src/main/java


Package Explorer
JUnit
edu.neu.coe.info6205.graphs.tunnels
edu.neu.coe.info6205.graphs.undirected
edu.neu.coe.info6205.greedy
edu.neu.coe.info6205.lab_1
edu.neu.coe.info6205.life.base
edu.neu.coe.info6205.life.library
edu.neu.coe.info6205.pq
edu.neu.coe.info6205.randomwalk
RandomWalk.java
edu.neu.coe.info6205.reduction
edu.neu.coe.info6205.runLengthEncoding
edu.neu.coe.info6205.sort
edu.neu.coe.info6205.sort.classic
edu.neu.coe.info6205.sort.counting
edu.neu.coe.info6205.sort.elementary
BubbleSort.java
InsertionSort.java
InsertionSortBasic.java
InsertionSortMSD.java
InsertionSortOpt.java
RandomSort.java
SelectionSort.java
ShellSort.java
Solution.java
SortMainMethod.java
edu.neu.coe.info6205.sort.hashCode
edu.neu.coe.info6205.sort.linearithmic
edu.neu.coe.info6205.sort.par
Main.java
ParSort.java
edu.neu.coe.info6205.symbolTable
edu.neu.coe.info6205.symbolTable.hashtable
edu.neu.coe.info6205.symbolTable.tree
edu.neu.coe.info6205.threesum
edu.neu.coe.info6205.union_find
Connections.java
HWQUPC_Solution.java
TypedUF_HWQUPC.java
TypedUF.java
UF_HWQUPC.java
UF.java
UFException.java
WQUPC.java
edu.neu.coe.info6205.util
Benchmark_Timer.java
Benchmark.java
Config.java
FastInverseSquareRoot.java
HWQUPC_Solut...
ParSort.java
Main.java
INFO6205/pom...
36
public class Main {
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolPara
Random random = new Random();
int[] array;
HashMap<String, ArrayList<Long>> timeMap = new HashMap<>();
ArrayList<Long> timeList = new ArrayList<>();
for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
System.out.println("Array size : " + arraySize);
array = new int[arraySize];
for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
ForkJoinPool myPool = new ForkJoinPool(threadCount);
System.out.println("Thread count is: " + threadCount);
for (int j = 50; j < 100; j += 5) {
ParSort.cutoff = 100 * (j + 1);
for (int i = 0; i < array.length; i++)
array[i] = random.nextInt(10000000);
Outline
edu.neu.coe.info6205.sort.par
Main
main(String[]) : void
processArgs(String[]) : void
processArg(String[]) : String[]
processCommand(String, Strin
setConfig(String, int) : void
configuration : Map<String, Int
Problems
Javadoc
Declaration
Search
Console
Terminal
Properties
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Array size : 300000
Thread count is: 2
cutoff: 5100 10times Time:138ms
cutoff: 5600 10times Time:170ms
cutoff: 6100 10times Time:140ms
cutoff: 6600 10times Time:138ms
cutoff: 7100 10times Time:163ms
cutoff: 7600 10times Time:138ms
cutoff: 8100 10times Time:168ms
cutoff: 8600 10times Time:149ms
cutoff: 9100 10times Time:148ms
cutoff: 9600 10times Time:156ms
Thread count is: 4
cutoff: 5100 10times Time:144ms
cutoff: 5600 10times Time:161ms
cutoff: 6100 10times Time:152ms
cutoff: 6600 10times Time:139ms
cutoff: 7100 10times Time:171ms
cutoff: 7600 10times Time:135ms
cutoff: 8100 10times Time:173ms
cutoff: 8600 10times Time:145ms
cutoff: 9100 10times Time:143ms
cutoff: 9600 10times Time:152ms
Thread count is: 8
cutoff: 5100 10times Time:152ms
edu.neu.coe.info6205.sort.par.Main.java - INFO6205/src/main/java



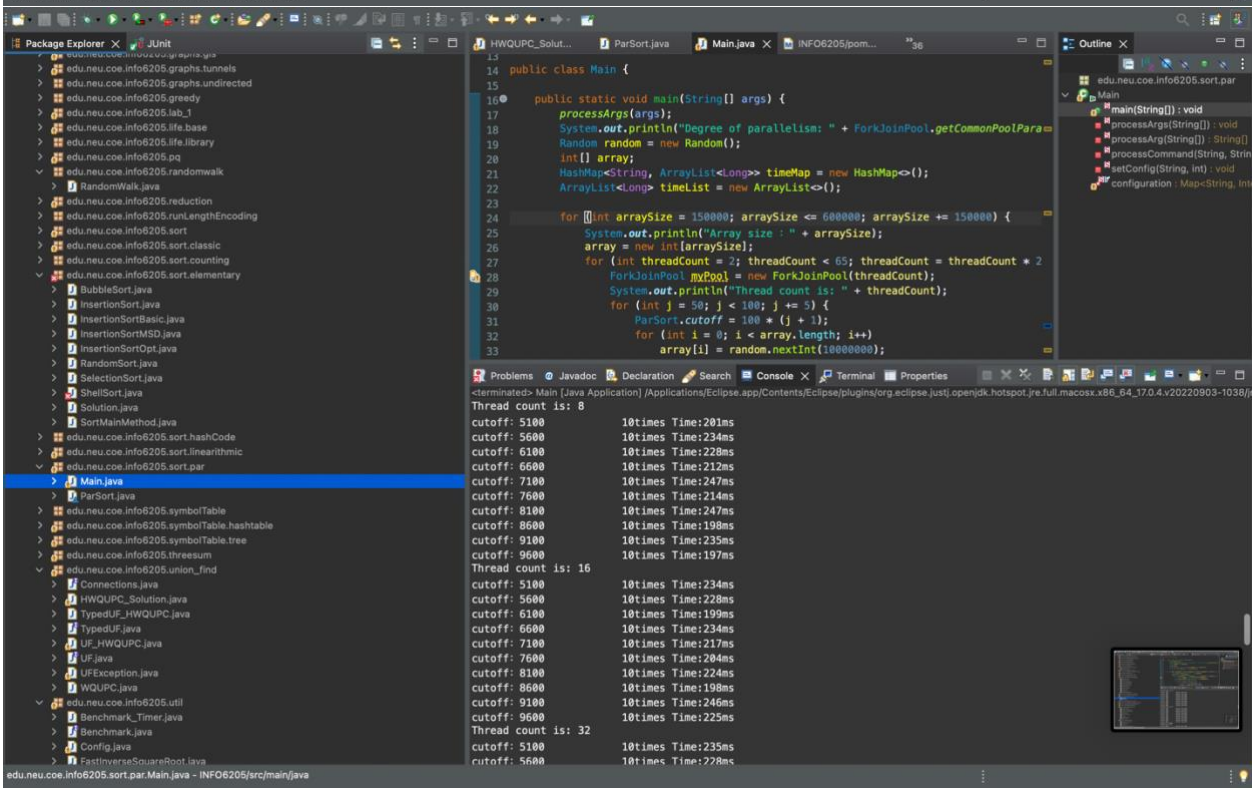
```
public class Main {

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

        for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
            System.out.println("Array size : " + arraySize);
            array = new int[arraySize];
            for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
                ForkJoinPool myPool = new ForkJoinPool(threadCount);
                System.out.println("Thread count is: " + threadCount);
                for (int j = 50; j < 100; j += 5) {
                    ParSort.cutoff = 100 * (j + 1);
                    for (int i = 0; i < array.length; i++)
                        array[i] = random.nextInt(10000000);
```

```
Thread count is: 8
cutoff: 5100            10times Time:152ms
cutoff: 5600            10times Time:143ms
cutoff: 6100            10times Time:176ms
cutoff: 6600            10times Time:150ms
cutoff: 7100            10times Time:162ms
cutoff: 7600            10times Time:159ms
cutoff: 8100            10times Time:148ms
cutoff: 8600            10times Time:172ms
cutoff: 9100            10times Time:143ms
cutoff: 9600            10times Time:159ms
Thread count is: 16
cutoff: 5100            10times Time:154ms
cutoff: 5600            10times Time:143ms
cutoff: 6100            10times Time:179ms
cutoff: 6600            10times Time:135ms
cutoff: 7100            10times Time:148ms
cutoff: 7600            10times Time:163ms
cutoff: 8100            10times Time:135ms
cutoff: 8600            10times Time:158ms
cutoff: 9100            10times Time:147ms
cutoff: 9600            10times Time:134ms
Thread count is: 32
cutoff: 5100            10times Time:177ms
cutoff: 5600            10times Time:141ms
cutoff: 6100            10times Time:168ms
cutoff: 6600            10times Time:136ms
cutoff: 7100            10times Time:140ms
cutoff: 7600            10times Time:172ms
cutoff: 8100            10times Time:145ms
cutoff: 8600            10times Time:144ms
cutoff: 9100            10times Time:144ms
cutoff: 9600            10times Time:193ms
Thread count is: 64
cutoff: 5100            10times Time:183ms
cutoff: 5600            10times Time:134ms
cutoff: 6100            10times Time:157ms
cutoff: 6600            10times Time:168ms
cutoff: 7100            10times Time:139ms
cutoff: 7600            10times Time:164ms
cutoff: 8100            10times Time:143ms
cutoff: 8600            10times Time:129ms
cutoff: 9100            10times Time:167ms
cutoff: 9600            10times Time:135ms
Array size : 450000
Thread count is: 2
cutoff: 5100            10times Time:216ms
```



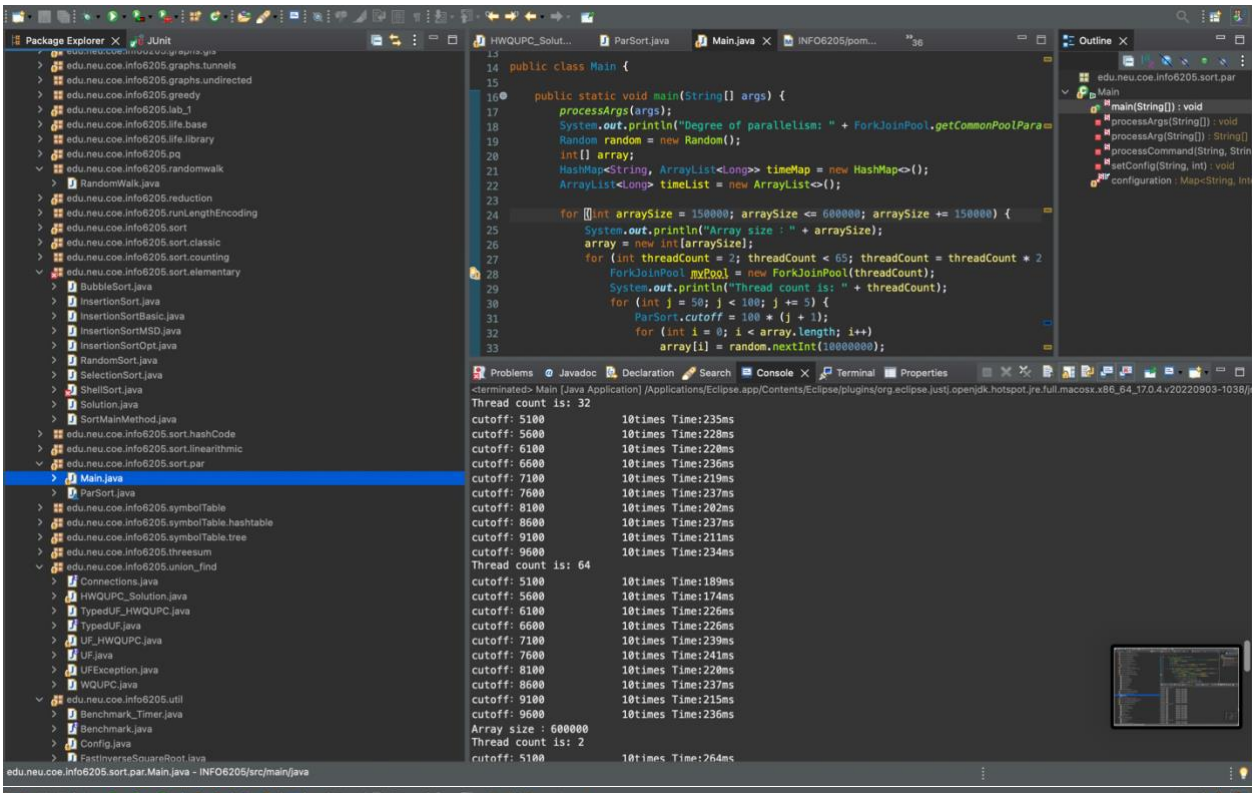

```
public class Main {

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

        for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
            System.out.println("Array size : " + arraySize);
            array = new int[arraySize];
            for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
                ForkJoinPool myPool = new ForkJoinPool(threadCount);
                System.out.println("Thread count is: " + threadCount);
                for (int j = 50; j < 100; j += 5) {
                    ParSort.cutoff = 100 * (j + 1);
                    for (int i = 0; i < array.length; i++)
                        array[i] = random.nextInt(10000000);
```

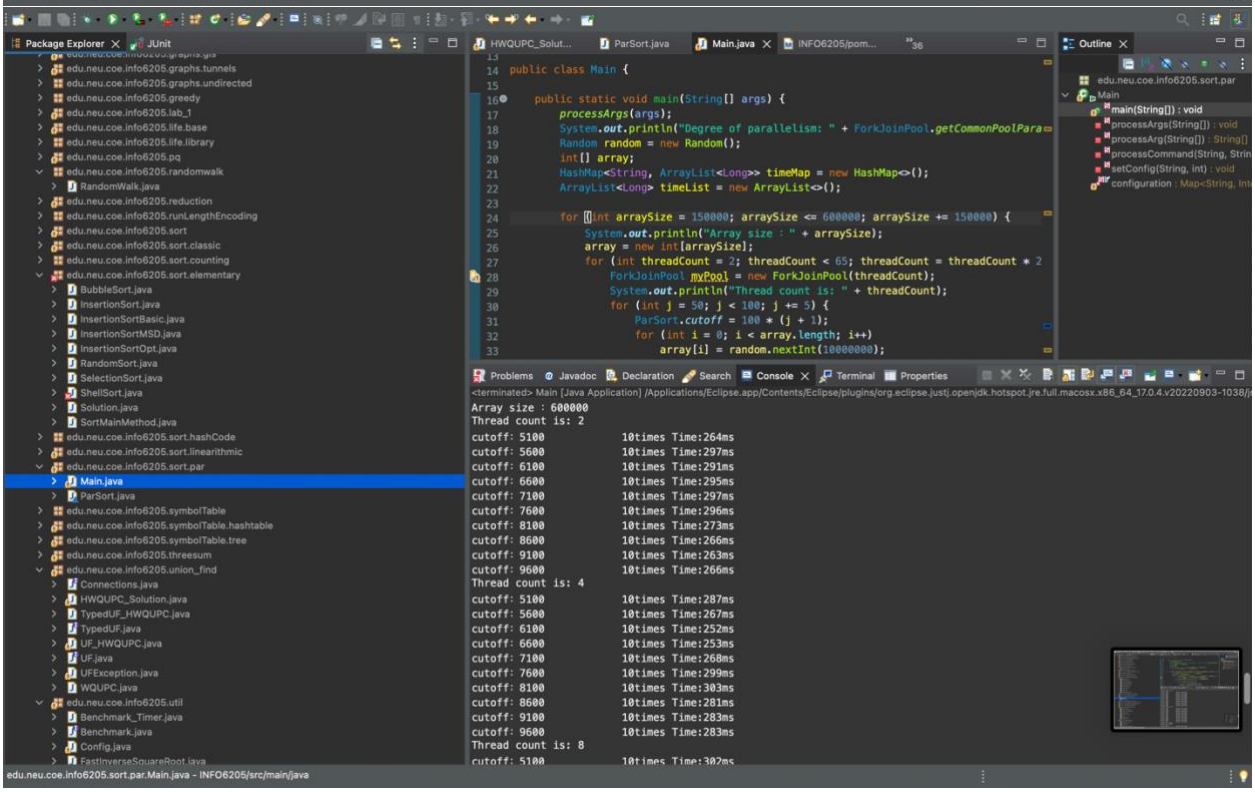
```
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Array size : 450000
Thread count is: 2
cutoff: 5100            10times Time:216ms
cutoff: 5600            10times Time:214ms
cutoff: 6100            10times Time:185ms
cutoff: 6600            10times Time:220ms
cutoff: 7100            10times Time:238ms
cutoff: 7600            10times Time:211ms
cutoff: 8100            10times Time:237ms
cutoff: 8600            10times Time:214ms
cutoff: 9100            10times Time:182ms
cutoff: 9600            10times Time:245ms
Thread count is: 4
cutoff: 5100            10times Time:208ms
cutoff: 5600            10times Time:231ms
cutoff: 6100            10times Time:213ms
cutoff: 6600            10times Time:208ms
cutoff: 7100            10times Time:236ms
cutoff: 7600            10times Time:247ms
cutoff: 8100            10times Time:211ms
cutoff: 8600            10times Time:212ms
cutoff: 9100            10times Time:202ms
cutoff: 9600            10times Time:246ms
Thread count is: 8
cutoff: 5100            10times Time:201ms
```

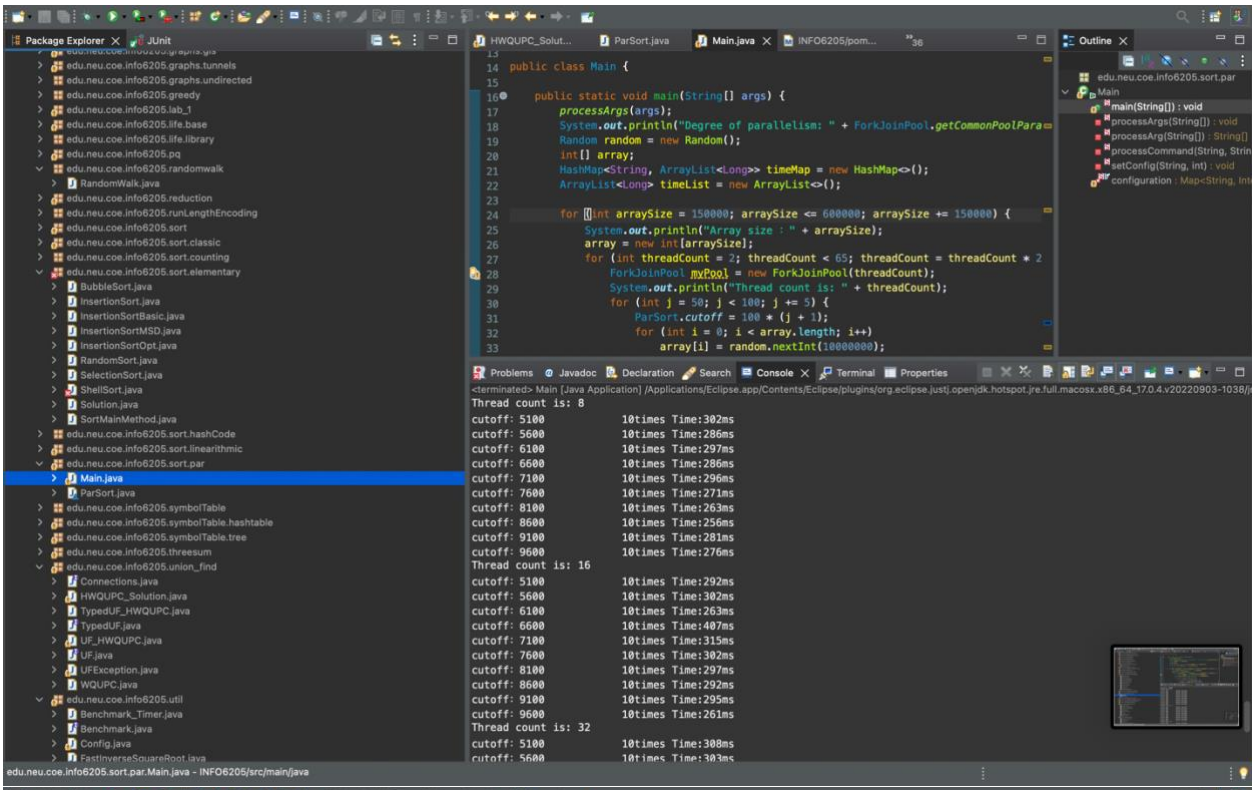


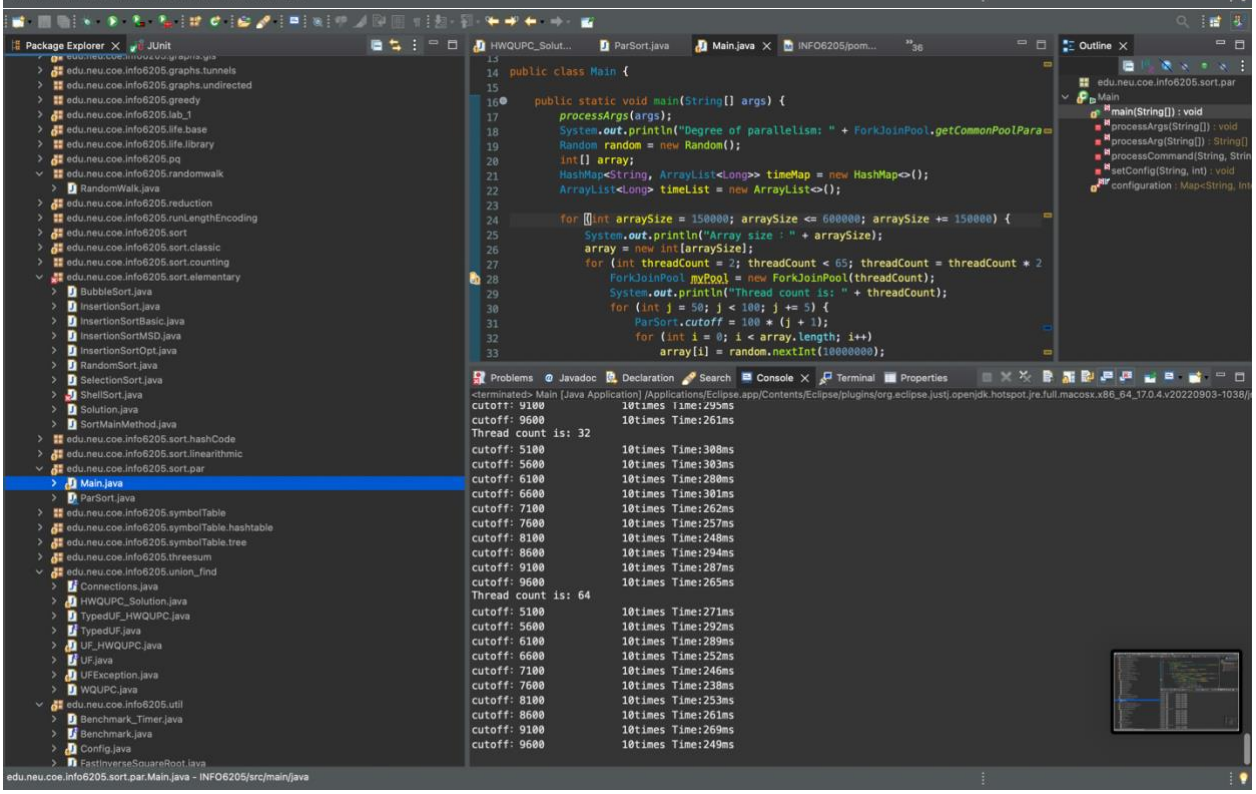

```
public class Main {

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

        for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
            System.out.println("Array size : " + arraySize);
            array = new int[arraySize];
            for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
                ForkJoinPool myPool = new ForkJoinPool(threadCount);
                System.out.println("Thread count is: " + threadCount);
                for (int j = 50; j < 100; j += 5) {
                    ParSort.cutoff = 100 * (j + 1);
                    for (int i = 0; i < array.length; i++)
                        array[i] = random.nextInt(10000000);
```

```
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Thread count is: 8
cutoff: 5100            10times Time:201ms
cutoff: 5600            10times Time:234ms
cutoff: 6100            10times Time:228ms
cutoff: 6600            10times Time:212ms
cutoff: 7100            10times Time:247ms
cutoff: 7600            10times Time:214ms
cutoff: 8100            10times Time:247ms
cutoff: 8600            10times Time:198ms
cutoff: 9100            10times Time:235ms
cutoff: 9600            10times Time:197ms
Thread count is: 16
cutoff: 5100            10times Time:234ms
cutoff: 5600            10times Time:228ms
cutoff: 6100            10times Time:199ms
cutoff: 6600            10times Time:234ms
cutoff: 7100            10times Time:217ms
cutoff: 7600            10times Time:204ms
cutoff: 8100            10times Time:224ms
cutoff: 8600            10times Time:198ms
cutoff: 9100            10times Time:246ms
cutoff: 9600            10times Time:225ms
Thread count is: 32
cutoff: 5100            10times Time:235ms
```

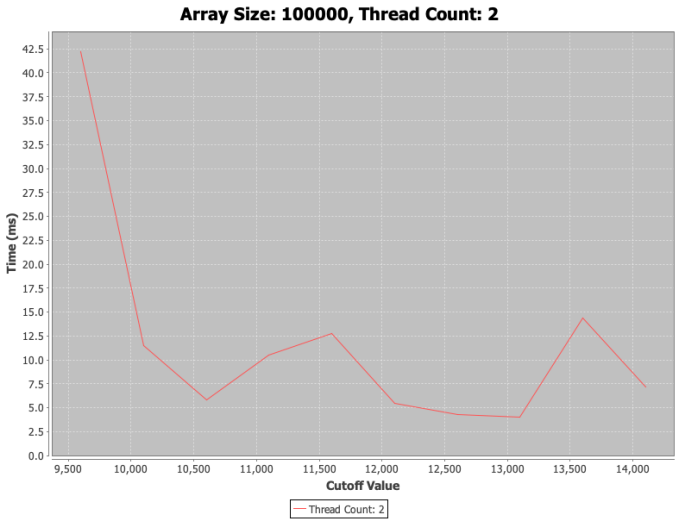

Package Explorer
JUnit
HWQUPC_Solut...
ParSort.java
Main.java
INFO6205/pom...
Outline
public class Main {
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolPara
Random random = new Random();
int[] array;
HashMap<String, ArrayList<Long>> timeMap = new HashMap<>();
ArrayList<Long> timeList = new ArrayList<>();
for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
System.out.println("Array size : " + arraySize);
array = new int[arraySize];
for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
ForkJoinPool myPool = new ForkJoinPool(threadCount);
System.out.println("Thread count is: " + threadCount);
for (int j = 50; j < 100; j += 5) {
ParSort.cutoff = 100 * (j + 1);
for (int i = 0; i < array.length; i++)
array[i] = random.nextInt(10000000);
Problems
Javadoc
Declaration
Search
Console
Terminal
Properties
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Thread count is: 32
cutoff: 5100 10times Time:235ms
cutoff: 5600 10times Time:228ms
cutoff: 6100 10times Time:220ms
cutoff: 6600 10times Time:236ms
cutoff: 7100 10times Time:219ms
cutoff: 7600 10times Time:237ms
cutoff: 8100 10times Time:202ms
cutoff: 8600 10times Time:237ms
cutoff: 9100 10times Time:211ms
cutoff: 9600 10times Time:234ms
Thread count is: 64
cutoff: 5100 10times Time:189ms
cutoff: 5600 10times Time:174ms
cutoff: 6100 10times Time:226ms
cutoff: 6600 10times Time:226ms
cutoff: 7100 10times Time:239ms
cutoff: 7600 10times Time:241ms
cutoff: 8100 10times Time:220ms
cutoff: 8600 10times Time:237ms
cutoff: 9100 10times Time:215ms
cutoff: 9600 10times Time:236ms
Array size : 600000
Thread count is: 2
edu.neu.coe.info6205.sort.par.Main.java - INFO6205/src/main/java

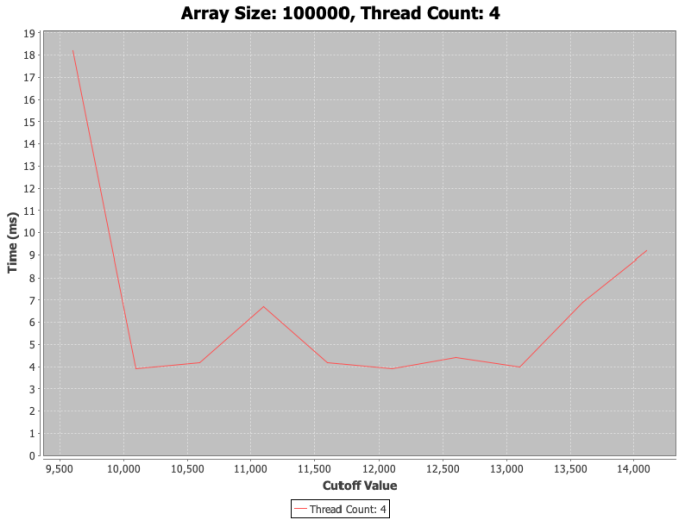

Package Explorer
JUnit
HWQUPC_Solut...
ParSort.java
Main.java
INFO6205/pom...
Outline
public class Main {
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolPara
Random random = new Random();
int[] array;
HashMap<String, ArrayList<Long>> timeMap = new HashMap<>();
ArrayList<Long> timeList = new ArrayList<>();
for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
System.out.println("Array size : " + arraySize);
array = new int[arraySize];
for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
ForkJoinPool myPool = new ForkJoinPool(threadCount);
System.out.println("Thread count is: " + threadCount);
for (int j = 50; j < 100; j += 5) {
ParSort.cutoff = 100 * (j + 1);
for (int i = 0; i < array.length; i++)
array[i] = random.nextInt(10000000);
Problems
Javadoc
Declaration
Search
Console
Terminal
Properties
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Array size : 600000
Thread count is: 2
cutoff: 5100 10times Time:264ms
cutoff: 5600 10times Time:297ms
cutoff: 6100 10times Time:291ms
cutoff: 6600 10times Time:295ms
cutoff: 7100 10times Time:297ms
cutoff: 7600 10times Time:296ms
cutoff: 8100 10times Time:273ms
cutoff: 8600 10times Time:266ms
cutoff: 9100 10times Time:263ms
cutoff: 9600 10times Time:266ms
Thread count is: 4
cutoff: 5100 10times Time:287ms
cutoff: 5600 10times Time:267ms
cutoff: 6100 10times Time:252ms
cutoff: 6600 10times Time:253ms
cutoff: 7100 10times Time:268ms
cutoff: 7600 10times Time:299ms
cutoff: 8100 10times Time:303ms
cutoff: 8600 10times Time:281ms
cutoff: 9100 10times Time:283ms
cutoff: 9600 10times Time:283ms
Thread count is: 8
edu.neu.coe.info6205.sort.par.Main.java - INFO6205/src/main/java

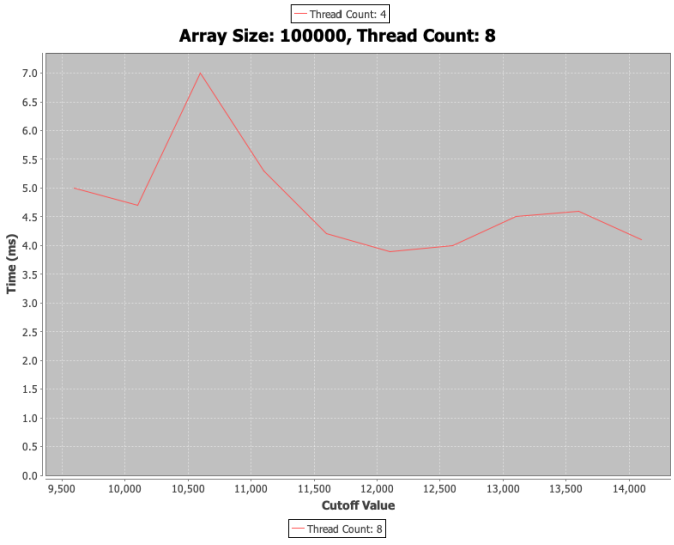

Package Explorer
JUnit
edu.neu.coe.info6205.graphs.tunnels
edu.neu.coe.info6205.graphs.undirected
edu.neu.coe.info6205.greedy
edu.neu.coe.info6205.lab_1
edu.neu.coe.info6205.life.base
edu.neu.coe.info6205.life.library
edu.neu.coe.info6205.pq
edu.neu.coe.info6205.randomwalk
RandomWalk.java
edu.neu.coe.info6205.reduction
edu.neu.coe.info6205.runLengthEncoding
edu.neu.coe.info6205.sort
edu.neu.coe.info6205.sort.classic
edu.neu.coe.info6205.sort.counting
edu.neu.coe.info6205.sort.elementary
BubbleSort.java
InsertionSort.java
InsertionSortBasic.java
InsertionSortMSD.java
InsertionSortOpt.java
RandomSort.java
SelectionSort.java
ShellSort.java
Solution.java
SortMainMethod.java
edu.neu.coe.info6205.sort.hashCode
edu.neu.coe.info6205.sort.linearithmic
edu.neu.coe.info6205.sort.par
Main.java
ParSort.java
edu.neu.coe.info6205.symbolTable
edu.neu.coe.info6205.symbolTable.hashtable
edu.neu.coe.info6205.symbolTable.tree
edu.neu.coe.info6205.threesum
edu.neu.coe.info6205.union_find
Connections.java
HWQUPC_Solution.java
TypedUF_HWQUPC.java
TypedUF.java
UF_HWQUPC.java
UF.java
UFException.java
WQUPC.java
edu.neu.coe.info6205.util
Benchmark_Timer.java
Benchmark.java
Config.java
FastInverseSquareRoot.java
HWQUPC_Solut...
ParSort.java
Main.java
INFO6205/pom...
36
public class Main {
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolPara
Random random = new Random();
int[] array;
HashMap<String, ArrayList<Long>> timeMap = new HashMap<>();
ArrayList<Long> timeList = new ArrayList<>();
for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
System.out.println("Array size : " + arraySize);
array = new int[arraySize];
for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
ForkJoinPool myPool = new ForkJoinPool(threadCount);
System.out.println("Thread count is: " + threadCount);
for (int j = 50; j < 100; j += 5) {
ParSort.cutoff = 100 * (j + 1);
for (int i = 0; i < array.length; i++)
array[i] = random.nextInt(10000000);
Outline
edu.neu.coe.info6205.sort.par
Main
main(String[]) : void
processArgs(String[]) : void
processArg(String[]) : String[]
processCommand(String, Strin
setConfig(String, int) : void
configuration : Map<String, Int
Problems
Javadoc
Declaration
Search
Console
Terminal
Properties
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
Thread count is: 8
cutoff: 5100 10times Time:302ms
cutoff: 5600 10times Time:286ms
cutoff: 6100 10times Time:297ms
cutoff: 6600 10times Time:286ms
cutoff: 7100 10times Time:296ms
cutoff: 7600 10times Time:271ms
cutoff: 8100 10times Time:263ms
cutoff: 8600 10times Time:256ms
cutoff: 9100 10times Time:281ms
cutoff: 9600 10times Time:276ms
Thread count is: 16
cutoff: 5100 10times Time:292ms
cutoff: 5600 10times Time:302ms
cutoff: 6100 10times Time:263ms
cutoff: 6600 10times Time:407ms
cutoff: 7100 10times Time:315ms
cutoff: 7600 10times Time:302ms
cutoff: 8100 10times Time:297ms
cutoff: 8600 10times Time:292ms
cutoff: 9100 10times Time:295ms
cutoff: 9600 10times Time:261ms
Thread count is: 32
cutoff: 5100 10times Time:308ms
edu.neu.coe.info6205.sort.par.Main.java - INFO6205/src/main/java

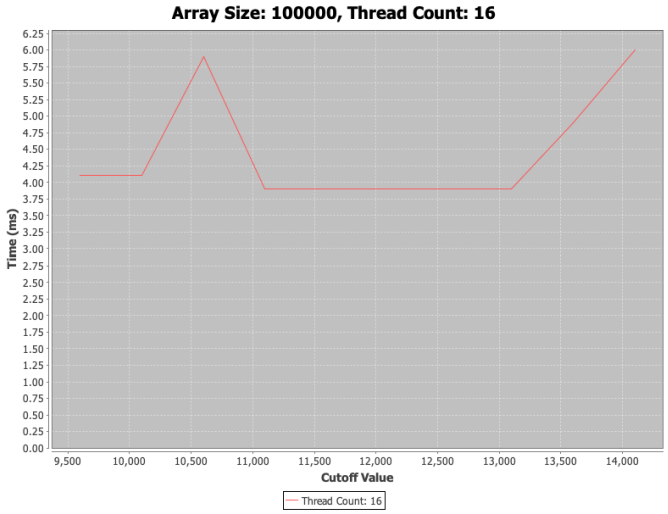

Package Explorer
JUnit
edu.neu.coe.info6205.graphs.tunnels
edu.neu.coe.info6205.graphs.undirected
edu.neu.coe.info6205.greedy
edu.neu.coe.info6205.lab_1
edu.neu.coe.info6205.life.base
edu.neu.coe.info6205.life.library
edu.neu.coe.info6205.pq
edu.neu.coe.info6205.randomwalk
RandomWalk.java
edu.neu.coe.info6205.reduction
edu.neu.coe.info6205.runLengthEncoding
edu.neu.coe.info6205.sort
edu.neu.coe.info6205.sort.classic
edu.neu.coe.info6205.sort.counting
edu.neu.coe.info6205.sort.elementary
BubbleSort.java
InsertionSort.java
InsertionSortBasic.java
InsertionSortMSD.java
InsertionSortOpt.java
RandomSort.java
SelectionSort.java
ShellSort.java
Solution.java
SortMainMethod.java
edu.neu.coe.info6205.sort.hashCode
edu.neu.coe.info6205.sort.linearithmic
edu.neu.coe.info6205.sort.par
Main.java
ParSort.java
edu.neu.coe.info6205.symbolTable
edu.neu.coe.info6205.symbolTable.hashtable
edu.neu.coe.info6205.symbolTable.tree
edu.neu.coe.info6205.threesum
edu.neu.coe.info6205.union_find
Connections.java
HWQUPC_Solution.java
TypedUF_HWQUPC.java
TypedUF.java
UF_HWQUPC.java
UF.java
UFException.java
WQUPC.java
edu.neu.coe.info6205.util
Benchmark_Timer.java
Benchmark.java
Config.java
FastInverseSquareRoot.java
HWQUPC_Solut...
ParSort.java
Main.java
INFO6205/pom...
36
public class Main {
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " + ForkJoinPool.getCommonPoolPara
Random random = new Random();
int[] array;
HashMap<String, ArrayList<Long>> timeMap = new HashMap<>();
ArrayList<Long> timeList = new ArrayList<>();
for (int arraySize = 150000; arraySize <= 600000; arraySize += 150000) {
System.out.println("Array size : " + arraySize);
array = new int[arraySize];
for (int threadCount = 2; threadCount < 65; threadCount = threadCount * 2
ForkJoinPool myPool = new ForkJoinPool(threadCount);
System.out.println("Thread count is: " + threadCount);
for (int j = 50; j < 100; j += 5) {
ParSort.cutoff = 100 * (j + 1);
for (int i = 0; i < array.length; i++)
array[i] = random.nextInt(10000000);
Outline
edu.neu.coe.info6205.sort.par
Main
main(String[]) : void
processArgs(String[]) : void
processArg(String[]) : String[]
processCommand(String, Strin
setConfig(String, int) : void
configuration : Map<String, Int
Problems
Javadoc
Declaration
Search
Console
Terminal
Properties
<terminated> Main [Java Application] /Applications/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.4.v20220903-1038/
cutoff: 9100 10times Time:295ms
cutoff: 9600 10times Time:261ms
Thread count is: 32
cutoff: 5100 10times Time:308ms
cutoff: 5600 10times Time:303ms
cutoff: 6100 10times Time:280ms
cutoff: 6600 10times Time:301ms
cutoff: 7100 10times Time:262ms
cutoff: 7600 10times Time:257ms
cutoff: 8100 10times Time:248ms
cutoff: 8600 10times Time:294ms
cutoff: 9100 10times Time:287ms
cutoff: 9600 10times Time:265ms
Thread count is: 64
cutoff: 5100 10times Time:271ms
cutoff: 5600 10times Time:292ms
cutoff: 6100 10times Time:289ms
cutoff: 6600 10times Time:252ms
cutoff: 7100 10times Time:246ms
cutoff: 7600 10times Time:238ms
cutoff: 8100 10times Time:253ms
cutoff: 8600 10times Time:261ms
cutoff: 9100 10times Time:269ms
cutoff: 9600 10times Time:249ms
edu.neu.coe.info6205.sort.par.Main.java - INFO6205/src/main/java

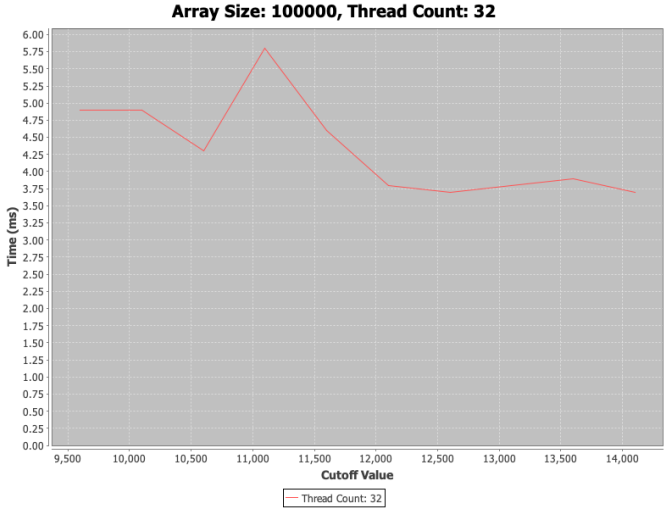
For ArraySize – 100000

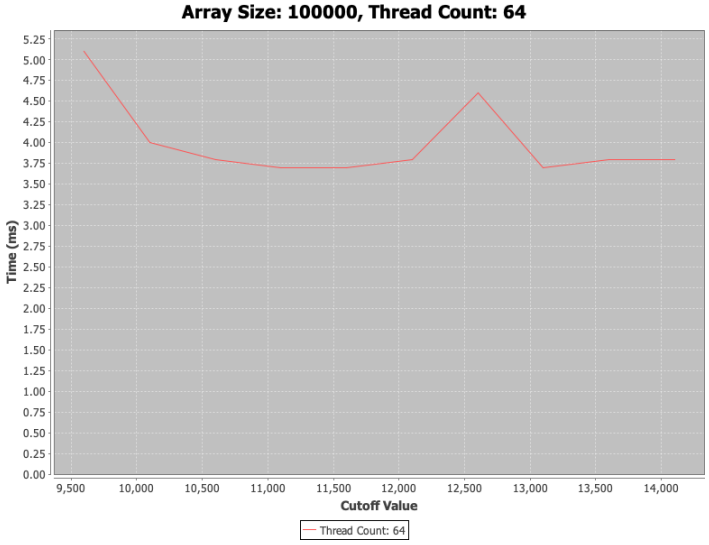




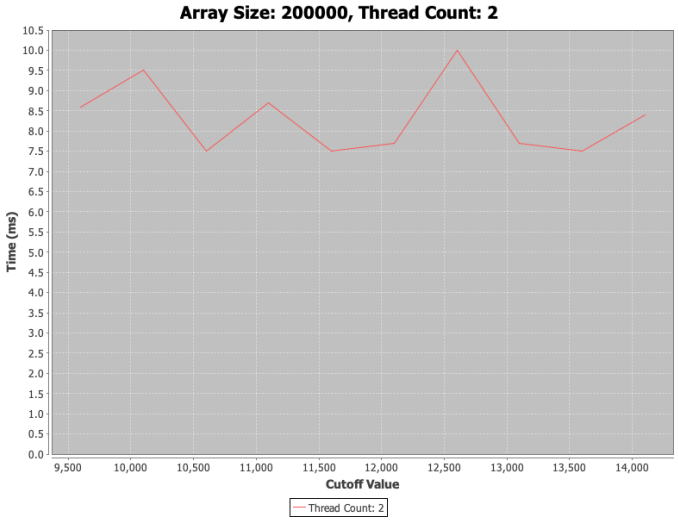


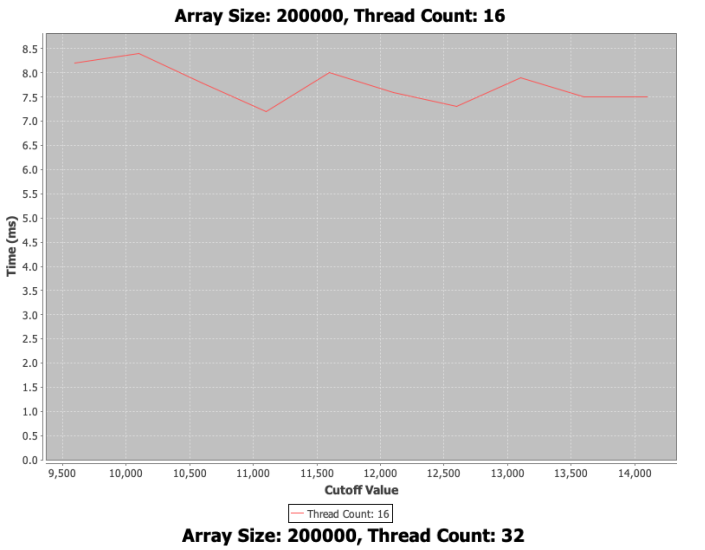

Array Size: 100000, Thread Count: 16
Time (ms)
6.25
6.00
5.75
5.50
5.25
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
9,500
10,000
10,500
11,000
11,500
12,000
12,500
13,000
13,500
14,000
Cutoff Value
Thread Count: 16

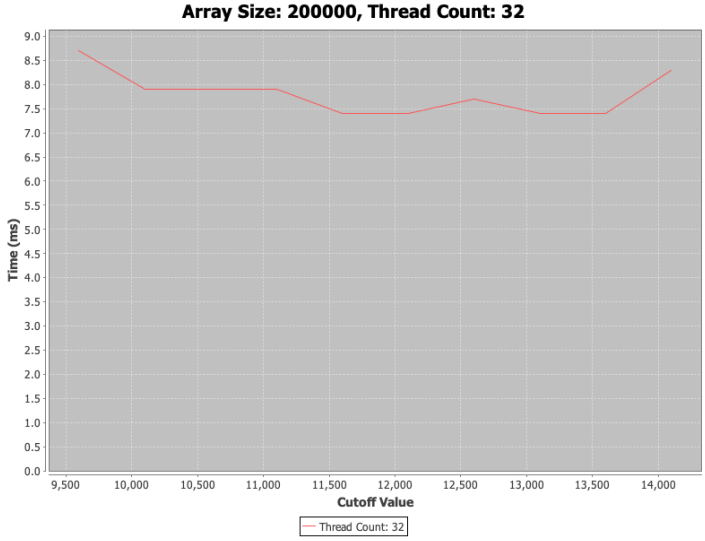

Array Size: 100000, Thread Count: 32
Time (ms)
6.00
5.75
5.50
5.25
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
9,500
10,000
10,500
11,000
11,500
12,000
12,500
13,000
13,500
14,000
Cutoff Value
Thread Count: 32

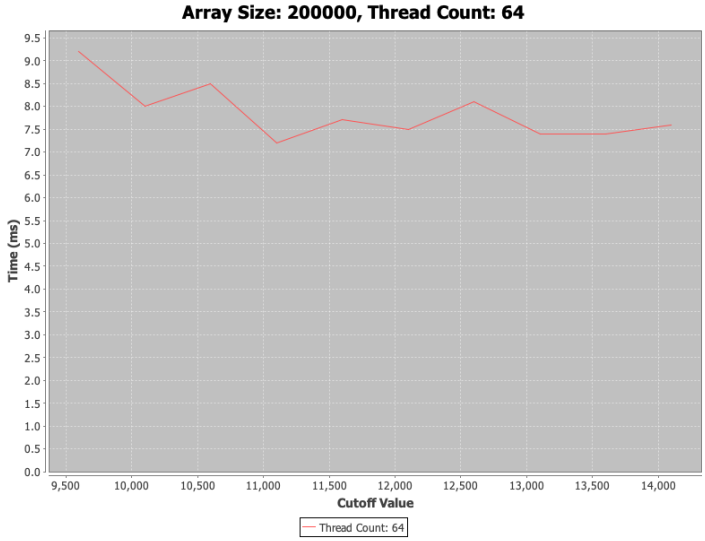

Array Size: 100000, Thread Count: 64
Time (ms)
5.25
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
9,500
10,000
10,500
11,000
11,500
12,000
12,500
13,000
13,500
14,000
Cutoff Value
Thread Count: 64

For ArraySize – 200000

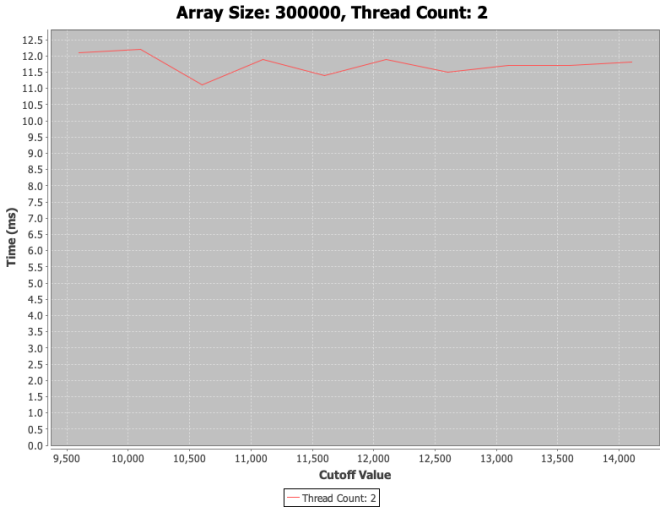


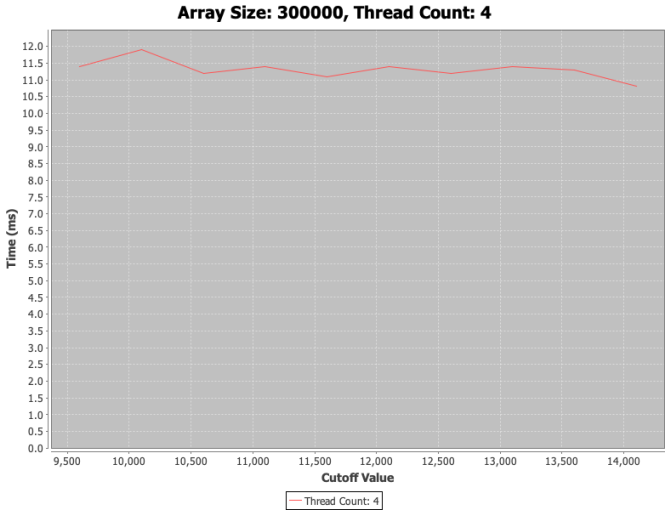

Array Size: 200000, Thread Count: 16
Time (ms)
Cutoff Value
Thread Count: 16

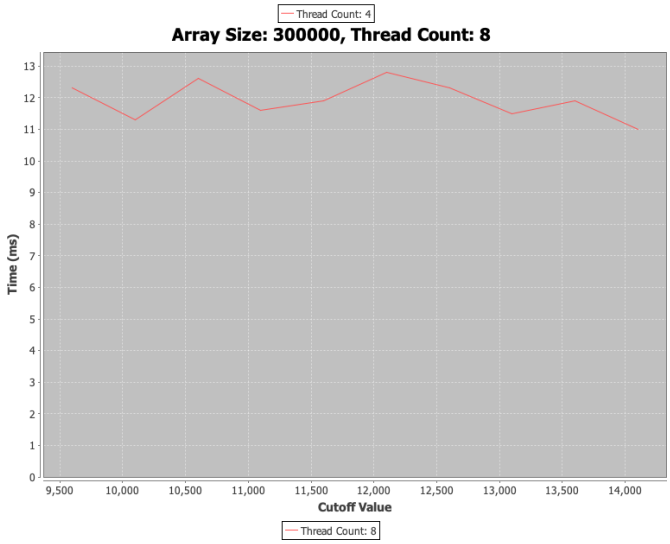

Array Size: 200000, Thread Count: 32
Time (ms)
Cutoff Value
Thread Count: 32


Array Size: 200000, Thread Count: 64
Time (ms)
Cutoff Value
Thread Count: 64

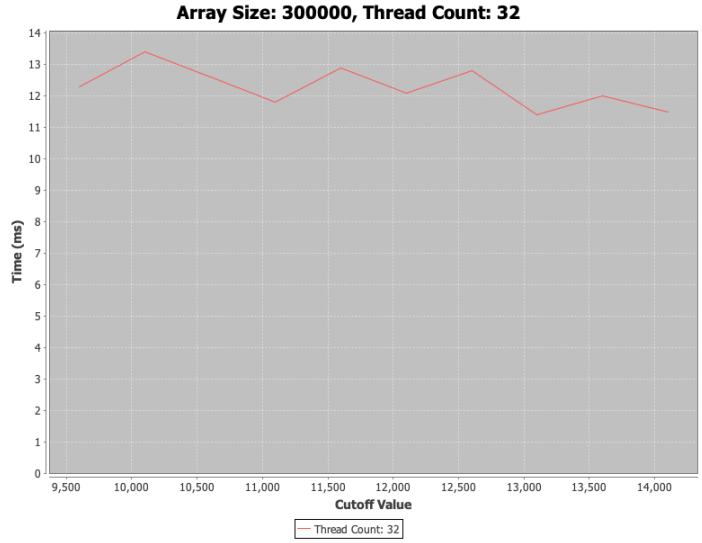
For ArraySize – 300000

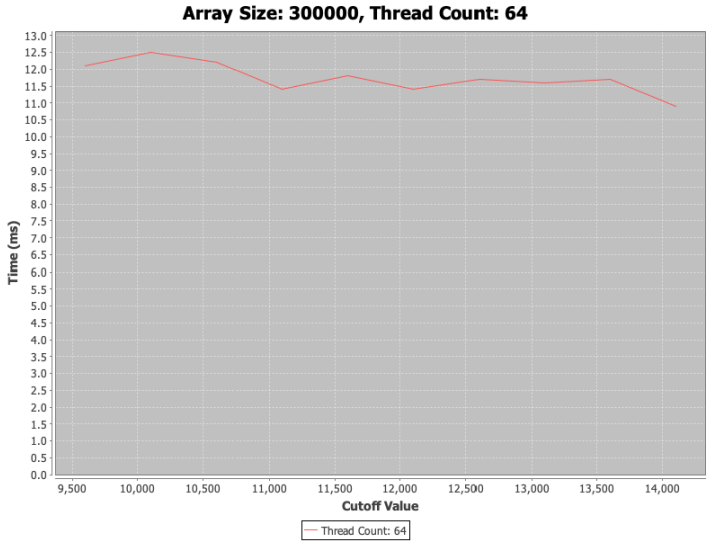







Array Size: 300000, Thread Count: 16
Time (ms)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
9,500
10,000
10,500
11,000
11,500
12,000
12,500
13,000
13,500
14,000
Cutoff Value
Thread Count: 16


Array Size: 300000, Thread Count: 32
Time (ms)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
9,500
10,000
10,500
11,000
11,500
12,000
12,500
13,000
13,500
14,000
Cutoff Value
Thread Count: 32


Array Size: 300000, Thread Count: 64
Time (ms)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
9.5
10.0
10.5
11.0
11.5
12.0
12.5
13.0
9,500
10,000
10,500
11,000
11,500
12,000
12,500
13,000
13,500
14,000
Cutoff Value
Thread Count: 64