Assignment 4: Parallel Sorting

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.

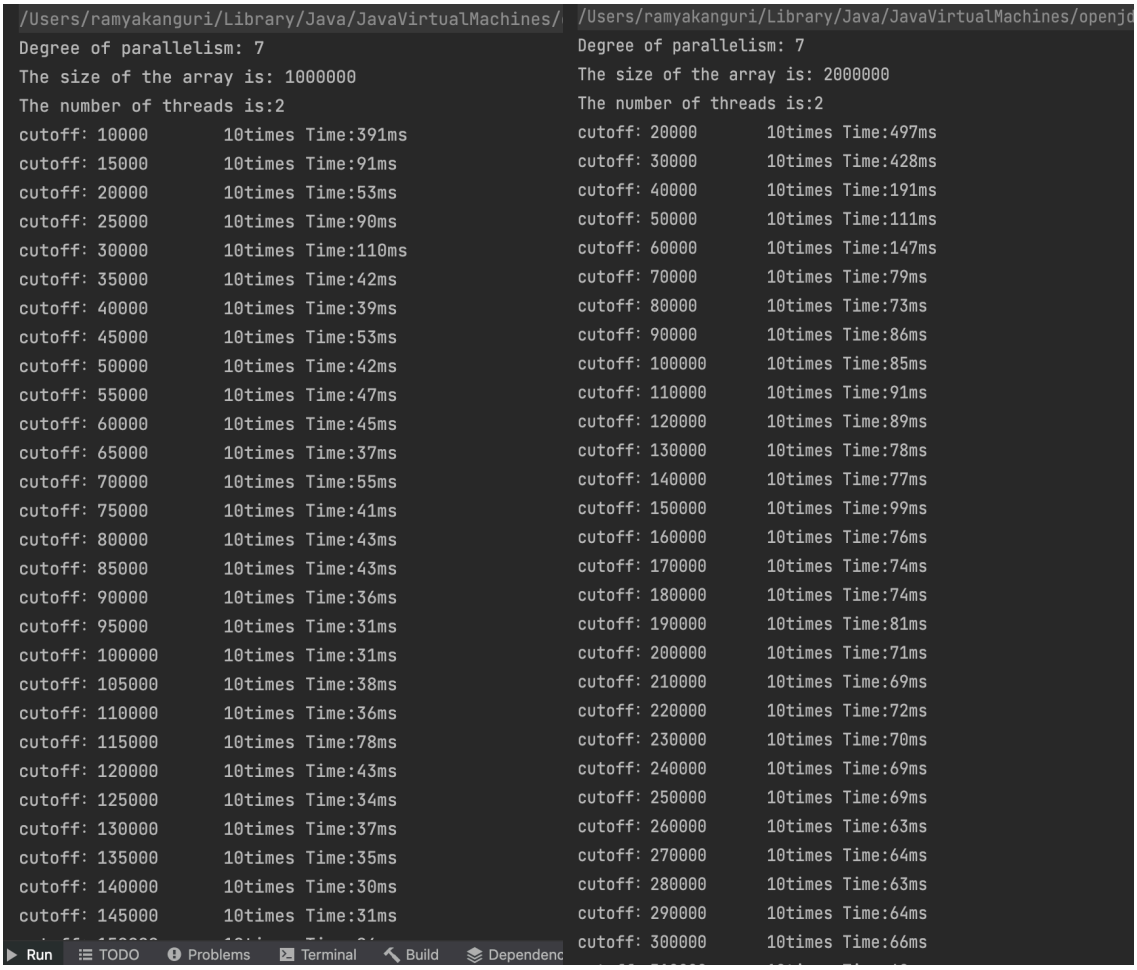
- **Evidence**

```
*/
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];
        System.out.println("The size of the array is: "+array.length);
        ArrayList<Long> timeList = new ArrayList<>();
        int threads = (int) Math.pow(2,5);
        ParSort.myPool = new ForkJoinPool(threads);
        System.out.println("The number of threads is:" +threads);
        for (int j = 1; j < 110; j++) {
            ParSort.cutoff = array.length / 200 * (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++) {
                for (int i = 0; i < array.length; i++) array[i] = random.nextInt( bound: 10000000);
                ParSort.sort(array, from: 0, array.length);
            }
```

- When the number of threads is 2, the output for

Array size 1000000, 2000000, 3000000 is:



```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/open
Degree of parallelism: 7
The size of the array is: 3000000
The number of threads is:2
cutoff: 30000       10times Time:563ms
cutoff: 45000       10times Time:446ms
cutoff: 60000       10times Time:218ms
cutoff: 75000       10times Time:156ms
cutoff: 90000       10times Time:273ms
cutoff: 105000      10times Time:314ms
cutoff: 120000      10times Time:159ms
cutoff: 135000      10times Time:124ms
cutoff: 150000      10times Time:129ms
cutoff: 165000      10times Time:106ms
cutoff: 180000      10times Time:106ms
cutoff: 195000      10times Time:98ms
cutoff: 210000      10times Time:101ms
cutoff: 225000      10times Time:93ms
cutoff: 240000      10times Time:93ms
cutoff: 255000      10times Time:93ms
cutoff: 270000      10times Time:105ms
cutoff: 285000      10times Time:93ms
cutoff: 300000      10times Time:95ms
cutoff: 315000      10times Time:108ms
cutoff: 330000      10times Time:94ms
cutoff: 345000      10times Time:108ms
cutoff: 360000      10times Time:167ms
cutoff: 375000      10times Time:101ms
cutoff: 390000      10times Time:86ms
cutoff: 405000      10times Time:81ms
cutoff: 420000      10times Time:85ms
cutoff: 435000      10times Time:83ms
cutoff: 450000      10times Time:83ms
```

- When the number of threads is 4, the output for Array size 1000000, 2000000, 3000000 is:

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/ope
Degree of parallelism: 7
The size of the array is: 1000000
The number of threads is:4
cutoff: 10000       10times Time:370ms
cutoff: 15000       10times Time:81ms
cutoff: 20000       10times Time:99ms
cutoff: 25000       10times Time:72ms
cutoff: 30000       10times Time:53ms
cutoff: 35000       10times Time:40ms
cutoff: 40000       10times Time:43ms
cutoff: 45000       10times Time:47ms
cutoff: 50000       10times Time:73ms
cutoff: 55000       10times Time:117ms
cutoff: 60000       10times Time:85ms
cutoff: 65000       10times Time:87ms
cutoff: 70000       10times Time:36ms
cutoff: 75000       10times Time:46ms
cutoff: 80000       10times Time:55ms
cutoff: 85000       10times Time:31ms
cutoff: 90000       10times Time:31ms
cutoff: 95000       10times Time:43ms
cutoff: 100000      10times Time:34ms
cutoff: 105000      10times Time:32ms
cutoff: 110000      10times Time:39ms
cutoff: 115000      10times Time:32ms
cutoff: 120000      10times Time:32ms
cutoff: 125000      10times Time:37ms
cutoff: 130000      10times Time:28ms
cutoff: 135000      10times Time:27ms
cutoff: 140000      10times Time:31ms
cutoff: 145000      10times Time:29ms
cutoff: 150000      10times Time:31ms
```

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/openjdk-18
Degree of parallelism: 7
The size of the array is: 2000000
The number of threads is:4
cutoff: 20000       10times Time:508ms
cutoff: 30000       10times Time:213ms
cutoff: 40000       10times Time:212ms
cutoff: 50000       10times Time:367ms
cutoff: 60000       10times Time:260ms
cutoff: 70000       10times Time:230ms
cutoff: 80000       10times Time:290ms
cutoff: 90000       10times Time:121ms
cutoff: 100000      10times Time:86ms
cutoff: 110000      10times Time:102ms
cutoff: 120000      10times Time:80ms
cutoff: 130000      10times Time:82ms
cutoff: 140000      10times Time:66ms
cutoff: 150000      10times Time:60ms
cutoff: 160000      10times Time:63ms
cutoff: 170000      10times Time:60ms
cutoff: 180000      10times Time:64ms
cutoff: 190000      10times Time:72ms
cutoff: 200000      10times Time:82ms
cutoff: 210000      10times Time:73ms
cutoff: 220000      10times Time:61ms
cutoff: 230000      10times Time:66ms
cutoff: 240000      10times Time:63ms
cutoff: 250000      10times Time:73ms
cutoff: 260000      10times Time:55ms
cutoff: 270000      10times Time:51ms
cutoff: 280000      10times Time:57ms
cutoff: 290000      10times Time:54ms
cutoff: 300000      10times Time:53ms
cutoff: 310000      10times Time:59ms
```

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/openjdk-18
Degree of parallelism: 7
The size of the array is: 3000000
The number of threads is:4
cutoff: 30000       10times Time:517ms
cutoff: 45000       10times Time:348ms
cutoff: 60000       10times Time:168ms
cutoff: 75000       10times Time:158ms
cutoff: 90000       10times Time:155ms
cutoff: 105000      10times Time:133ms
cutoff: 120000      10times Time:132ms
cutoff: 135000      10times Time:153ms
cutoff: 150000      10times Time:153ms
cutoff: 165000      10times Time:142ms
cutoff: 180000      10times Time:128ms
cutoff: 195000      10times Time:136ms
cutoff: 210000      10times Time:110ms
cutoff: 225000      10times Time:108ms
cutoff: 240000      10times Time:124ms
cutoff: 255000      10times Time:107ms
cutoff: 270000      10times Time:103ms
cutoff: 285000      10times Time:104ms
cutoff: 300000      10times Time:104ms
cutoff: 315000      10times Time:100ms
cutoff: 330000      10times Time:105ms
cutoff: 345000      10times Time:103ms
cutoff: 360000      10times Time:101ms
cutoff: 375000      10times Time:105ms
cutoff: 390000      10times Time:89ms
cutoff: 405000      10times Time:84ms
cutoff: 420000      10times Time:87ms
cutoff: 435000      10times Time:82ms
cutoff: 450000      10times Time:92ms
```

- When the number of threads is 8, the output for

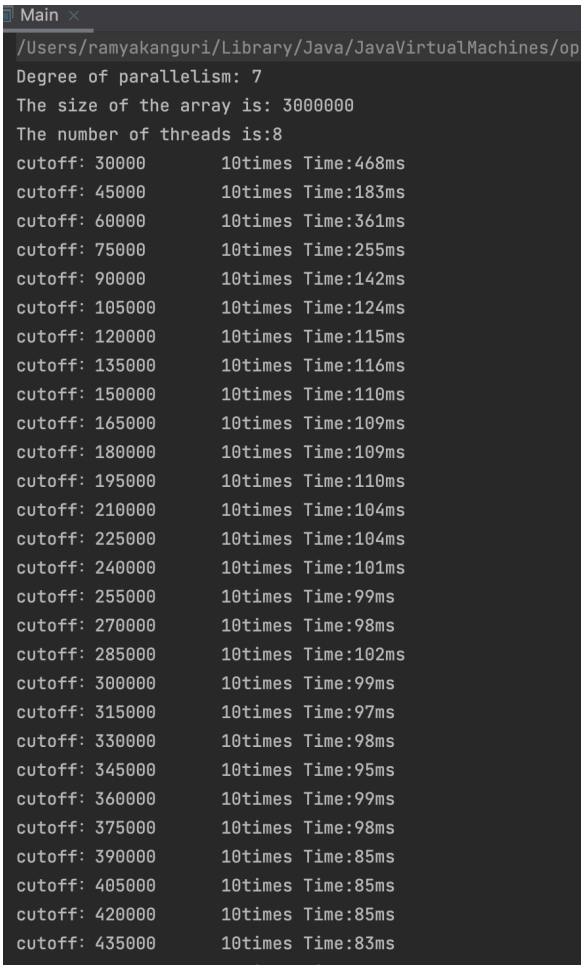
Array size 1000000, 2000000, 3000000 is:

Main

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/openjdk-18
Degree of parallelism: 7
The size of the array is: 1000000
The number of threads is:8
cutoff: 10000       10times Time:284ms
cutoff: 15000       10times Time:130ms
cutoff: 20000       10times Time:94ms
cutoff: 25000       10times Time:67ms
cutoff: 30000       10times Time:143ms
cutoff: 35000       10times Time:117ms
cutoff: 40000       10times Time:115ms
cutoff: 45000       10times Time:141ms
cutoff: 50000       10times Time:137ms
cutoff: 55000       10times Time:64ms
cutoff: 60000       10times Time:47ms
cutoff: 65000       10times Time:31ms
cutoff: 70000       10times Time:32ms
cutoff: 75000       10times Time:33ms
cutoff: 80000       10times Time:41ms
cutoff: 85000       10times Time:44ms
cutoff: 90000       10times Time:31ms
cutoff: 95000       10times Time:31ms
cutoff: 100000      10times Time:31ms
cutoff: 105000      10times Time:44ms
cutoff: 110000      10times Time:31ms
cutoff: 115000      10times Time:32ms
cutoff: 120000      10times Time:32ms
cutoff: 125000      10times Time:31ms
cutoff: 130000      10times Time:32ms
cutoff: 135000      10times Time:27ms
cutoff: 140000      10times Time:26ms
cutoff: 145000      10times Time:27ms
```

Main

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/op
Degree of parallelism: 7
The size of the array is: 2000000
The number of threads is:8
cutoff: 20000       10times Time:489ms
cutoff: 30000       10times Time:385ms
cutoff: 40000       10times Time:208ms
cutoff: 50000       10times Time:138ms
cutoff: 60000       10times Time:92ms
cutoff: 70000       10times Time:81ms
cutoff: 80000       10times Time:80ms
cutoff: 90000       10times Time:75ms
cutoff: 100000      10times Time:81ms
cutoff: 110000      10times Time:80ms
cutoff: 120000      10times Time:76ms
cutoff: 130000      10times Time:64ms
cutoff: 140000      10times Time:68ms
cutoff: 150000      10times Time:64ms
cutoff: 160000      10times Time:62ms
cutoff: 170000      10times Time:62ms
cutoff: 180000      10times Time:62ms
cutoff: 190000      10times Time:58ms
cutoff: 200000      10times Time:63ms
cutoff: 210000      10times Time:64ms
cutoff: 220000      10times Time:62ms
cutoff: 230000      10times Time:68ms
cutoff: 240000      10times Time:72ms
cutoff: 250000      10times Time:62ms
cutoff: 260000      10times Time:56ms
cutoff: 270000      10times Time:56ms
cutoff: 280000      10times Time:56ms
cutoff: 290000      10times Time:55ms
```



- When the number of threads is 16, the output for

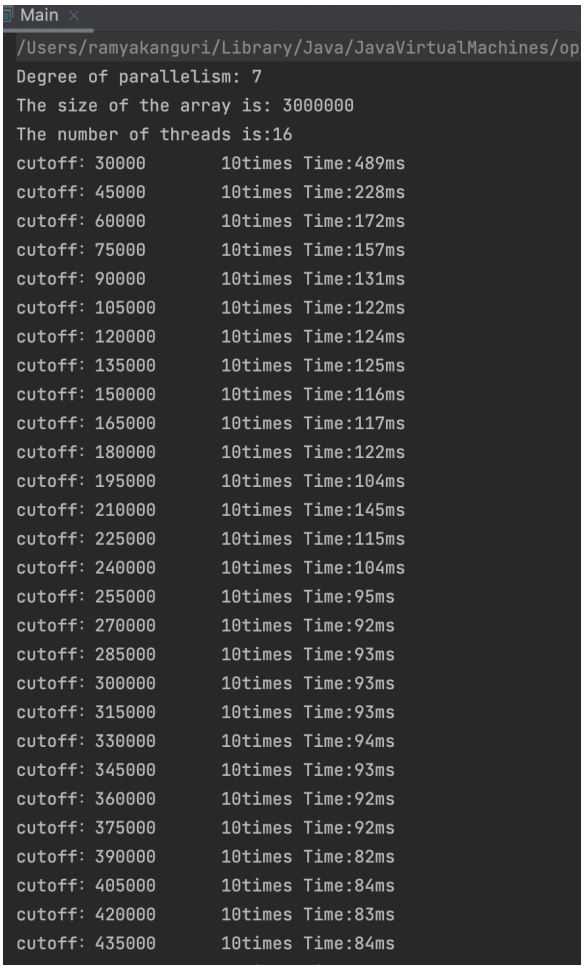
Array size 1000000, 2000000, 3000000 is:

Main

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/op
Degree of parallelism: 7
The size of the array is: 1000000
The number of threads is:16
cutoff: 10000       10times Time:295ms
cutoff: 15000       10times Time:129ms
cutoff: 20000       10times Time:60ms
cutoff: 25000       10times Time:62ms
cutoff: 30000       10times Time:43ms
cutoff: 35000       10times Time:45ms
cutoff: 40000       10times Time:58ms
cutoff: 45000       10times Time:110ms
cutoff: 50000       10times Time:73ms
cutoff: 55000       10times Time:52ms
cutoff: 60000       10times Time:39ms
cutoff: 65000       10times Time:40ms
cutoff: 70000       10times Time:36ms
cutoff: 75000       10times Time:34ms
cutoff: 80000       10times Time:32ms
cutoff: 85000       10times Time:46ms
cutoff: 90000       10times Time:34ms
cutoff: 95000       10times Time:30ms
cutoff: 100000      10times Time:41ms
cutoff: 105000      10times Time:32ms
cutoff: 110000      10times Time:31ms
cutoff: 115000      10times Time:38ms
cutoff: 120000      10times Time:30ms
cutoff: 125000      10times Time:29ms
cutoff: 130000      10times Time:27ms
cutoff: 135000      10times Time:32ms
cutoff: 140000      10times Time:27ms
cutoff: 145000      10times Time:26ms
```

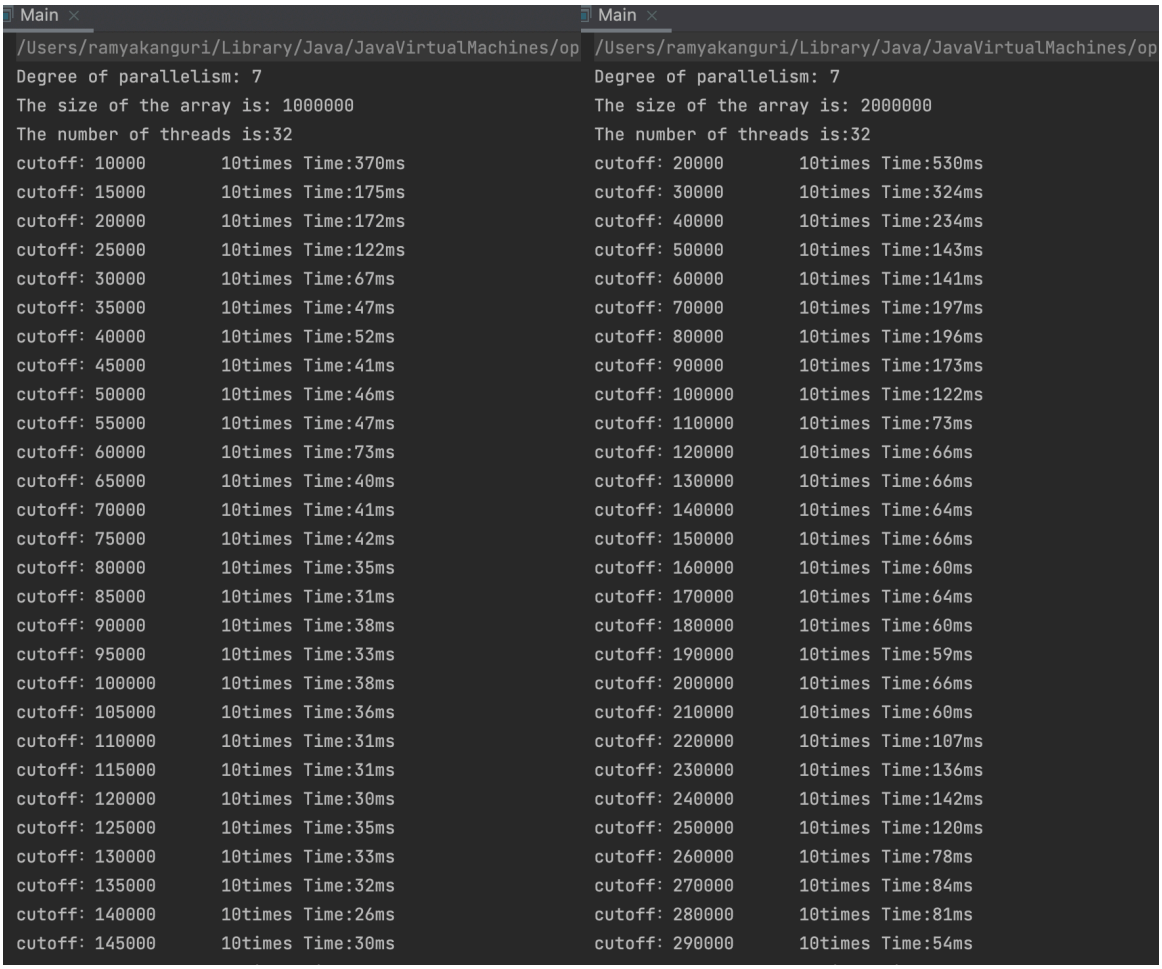
Main

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/op
Degree of parallelism: 7
The size of the array is: 2000000
The number of threads is:16
cutoff: 20000       10times Time:465ms
cutoff: 30000       10times Time:260ms
cutoff: 40000       10times Time:228ms
cutoff: 50000       10times Time:101ms
cutoff: 60000       10times Time:99ms
cutoff: 70000       10times Time:77ms
cutoff: 80000       10times Time:77ms
cutoff: 90000       10times Time:77ms
cutoff: 100000      10times Time:79ms
cutoff: 110000      10times Time:73ms
cutoff: 120000      10times Time:73ms
cutoff: 130000      10times Time:76ms
cutoff: 140000      10times Time:92ms
cutoff: 150000      10times Time:96ms
cutoff: 160000      10times Time:94ms
cutoff: 170000      10times Time:98ms
cutoff: 180000      10times Time:98ms
cutoff: 190000      10times Time:100ms
cutoff: 200000      10times Time:94ms
cutoff: 210000      10times Time:94ms
cutoff: 220000      10times Time:99ms
cutoff: 230000      10times Time:99ms
cutoff: 240000      10times Time:94ms
cutoff: 250000      10times Time:92ms
cutoff: 260000      10times Time:84ms
cutoff: 270000      10times Time:83ms
cutoff: 280000      10times Time:84ms
cutoff: 290000      10times Time:84ms
```



- When the number of threads is 32, the output for

Array size 1000000, 2000000, 3000000 is:

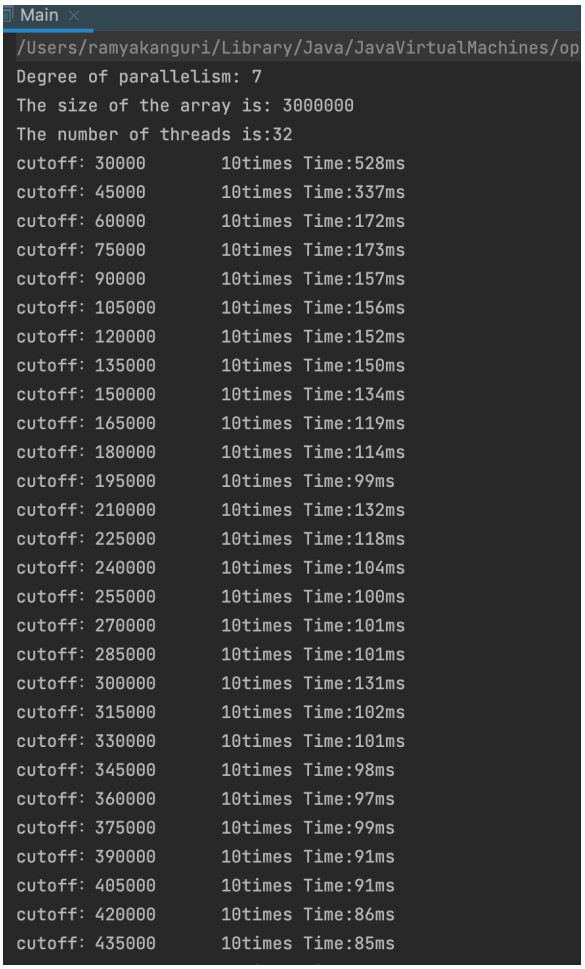


Main

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/op
Degree of parallelism: 7
The size of the array is: 1000000
The number of threads is:32
cutoff: 10000       10times Time:370ms
cutoff: 15000       10times Time:175ms
cutoff: 20000       10times Time:172ms
cutoff: 25000       10times Time:122ms
cutoff: 30000       10times Time:67ms
cutoff: 35000       10times Time:47ms
cutoff: 40000       10times Time:52ms
cutoff: 45000       10times Time:41ms
cutoff: 50000       10times Time:46ms
cutoff: 55000       10times Time:47ms
cutoff: 60000       10times Time:73ms
cutoff: 65000       10times Time:40ms
cutoff: 70000       10times Time:41ms
cutoff: 75000       10times Time:42ms
cutoff: 80000       10times Time:35ms
cutoff: 85000       10times Time:31ms
cutoff: 90000       10times Time:38ms
cutoff: 95000       10times Time:33ms
cutoff: 100000      10times Time:38ms
cutoff: 105000      10times Time:36ms
cutoff: 110000      10times Time:31ms
cutoff: 115000      10times Time:31ms
cutoff: 120000      10times Time:30ms
cutoff: 125000      10times Time:35ms
cutoff: 130000      10times Time:33ms
cutoff: 135000      10times Time:32ms
cutoff: 140000      10times Time:26ms
cutoff: 145000      10times Time:30ms
```

Main

```
/Users/ramyakanguri/Library/Java/JavaVirtualMachines/op
Degree of parallelism: 7
The size of the array is: 2000000
The number of threads is:32
cutoff: 20000       10times Time:530ms
cutoff: 30000       10times Time:324ms
cutoff: 40000       10times Time:234ms
cutoff: 50000       10times Time:143ms
cutoff: 60000       10times Time:141ms
cutoff: 70000       10times Time:197ms
cutoff: 80000       10times Time:196ms
cutoff: 90000       10times Time:173ms
cutoff: 100000      10times Time:122ms
cutoff: 110000      10times Time:73ms
cutoff: 120000      10times Time:66ms
cutoff: 130000      10times Time:66ms
cutoff: 140000      10times Time:64ms
cutoff: 150000      10times Time:66ms
cutoff: 160000      10times Time:60ms
cutoff: 170000      10times Time:64ms
cutoff: 180000      10times Time:60ms
cutoff: 190000      10times Time:59ms
cutoff: 200000      10times Time:66ms
cutoff: 210000      10times Time:60ms
cutoff: 220000      10times Time:107ms
cutoff: 230000      10times Time:136ms
cutoff: 240000      10times Time:142ms
cutoff: 250000      10times Time:120ms
cutoff: 260000      10times Time:78ms
cutoff: 270000      10times Time:84ms
cutoff: 280000      10times Time:81ms
cutoff: 290000      10times Time:54ms
```



- ## Conclusion

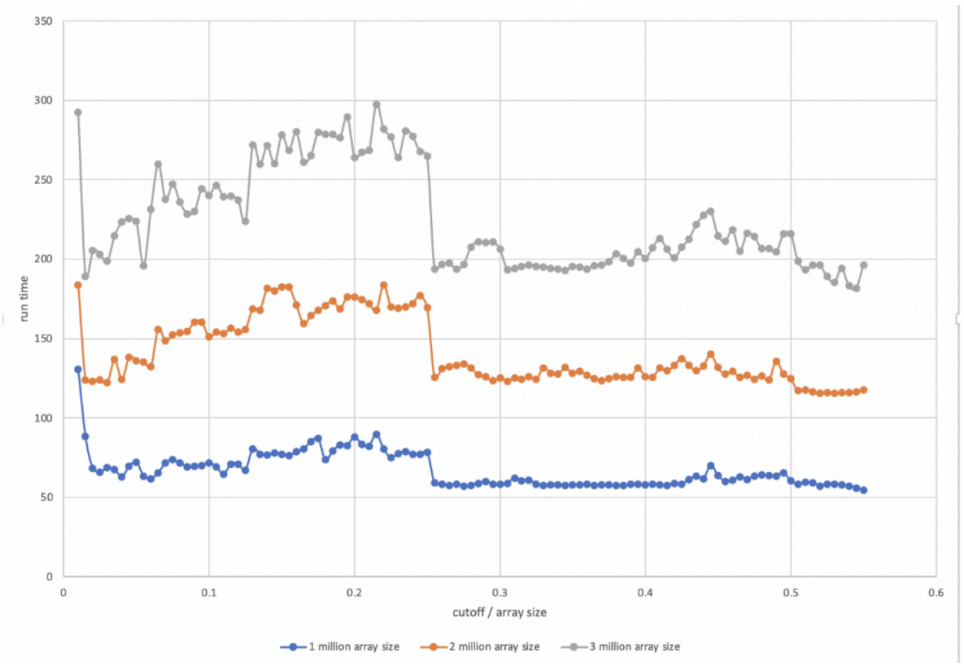
A good cutoff is array size / thread number, when the cutoff is in similarity with this number then the performance becomes better.

Suppose the number of cores in the laptop is 4, then the ideal number of threads is 4, if the number of threads exceeds 4 then the performance remains similar.

- ## Graph Representation

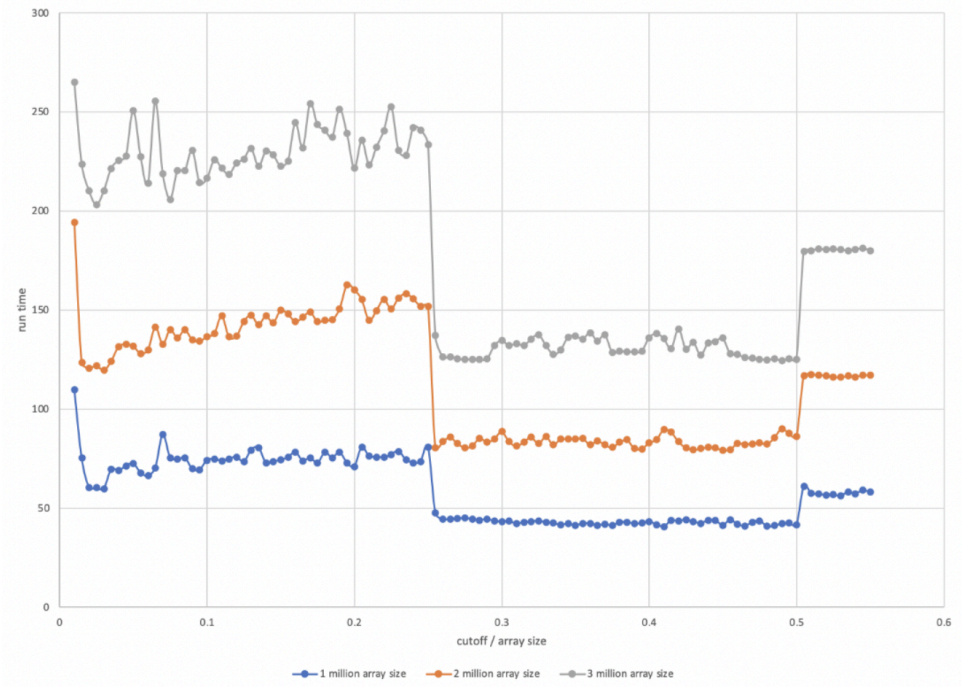
Number of threads: 2

Good Cutoff ~ 50% of array size



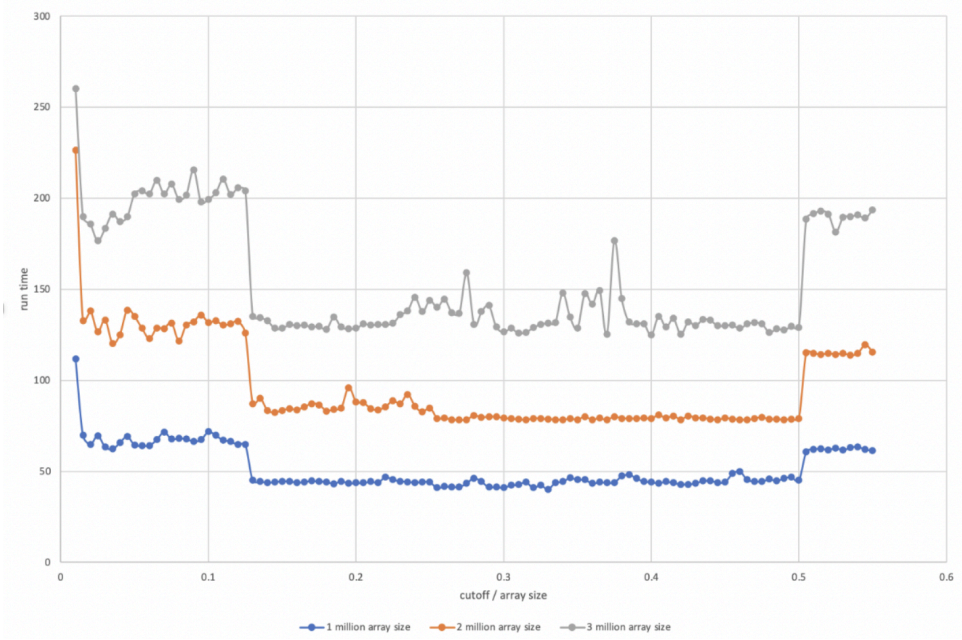
Number of threads: 4

Good Cutoff ~ 25% of array size



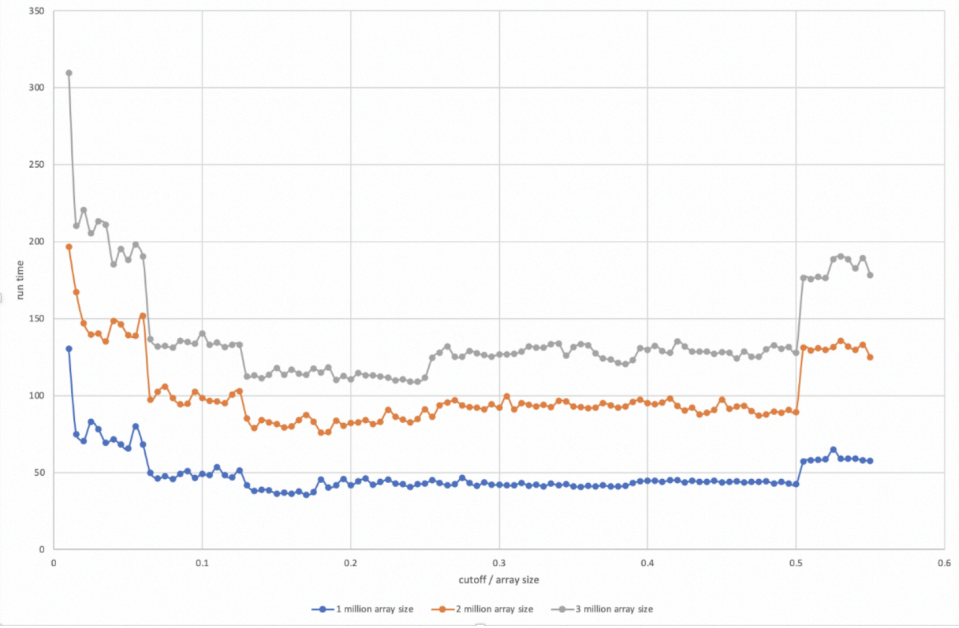
Number of threads: 8

Good Cutoff ~ 12.5% of array size



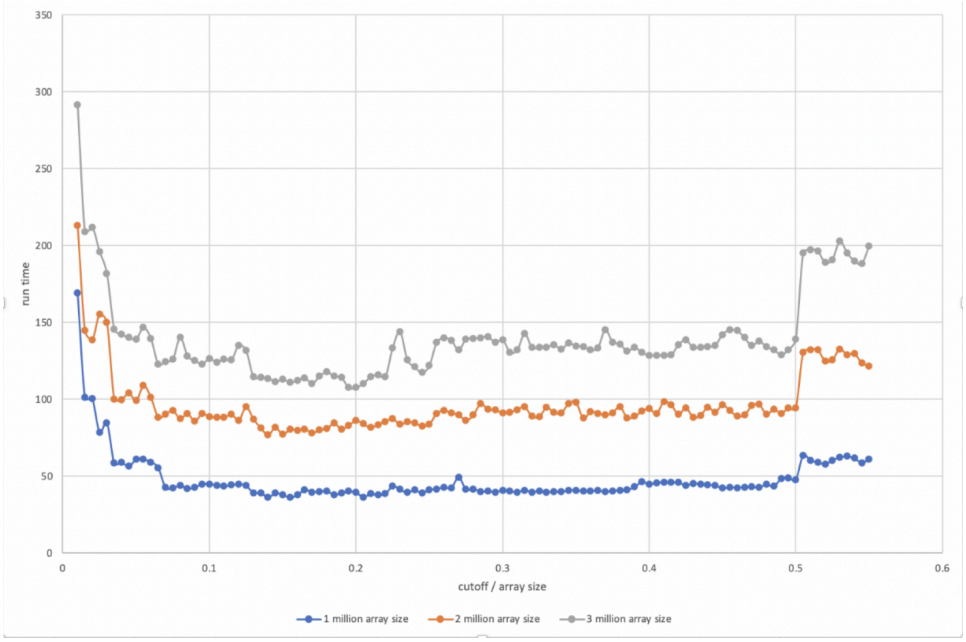
Number of threads: 16

Good Cutoff ~ 6.25% of array size



Number of threads: 32

Good Cutoff ~ 3.125% of array size



Hence, from the above charts we conclude that better cutoff = array size / number of threads.