Apoorva Kethamaranahalli: 001090928

Program Structures & Algorithms

Fall 2021

Assignment No. 5

Task

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.

Relationship Conclusion:

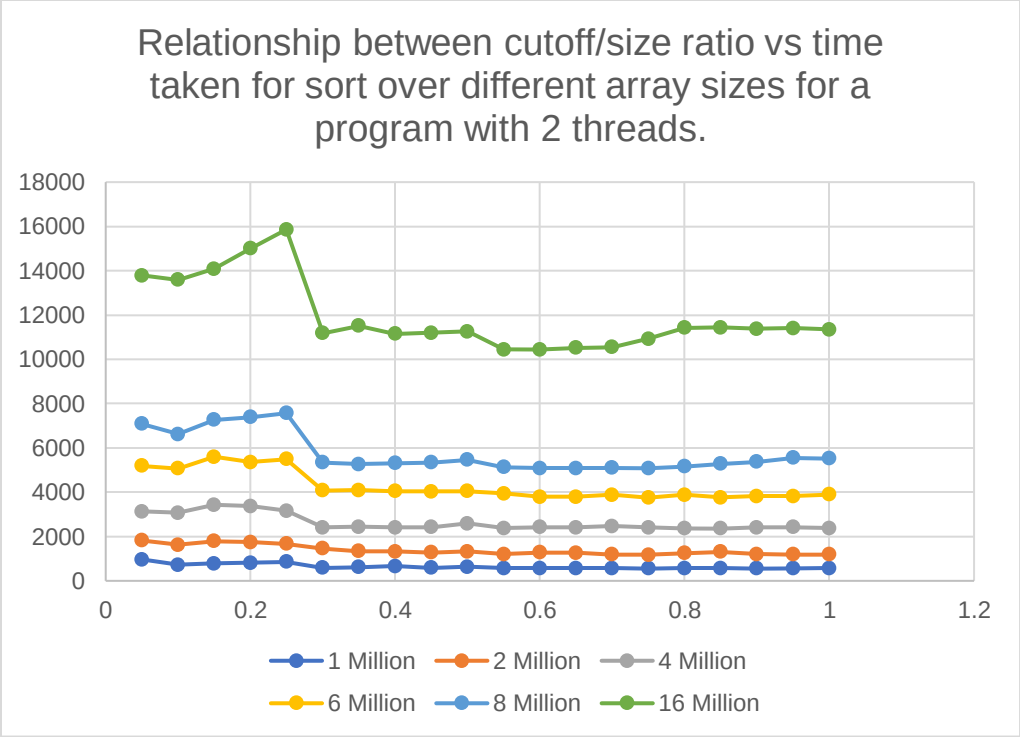
1. As per the cutoff ratio (cutoff/size of the array), we see that irrespective of the array sizes, and the number of threads available for parallel sorting, there is a range of cutoff ratio between 0.3 and 0.55 for which parallel sort can be an optimal solution
2. As per performing analysis on different sizes of arrays (2 Million and 16 Million), for a number of different threads ranging from 2 to 16, we see that 6 threads works as a best case scenario

Evidence to support the conclusion: (Graphical Representation)

1. Finding the optimal cutoff value, for different sizes of input array(while keeping the number of threads as a constant)

    - Threads : 2 [Refer fig: 2.0 to 2.4 in Output Screenshots]
      Array sizes varied from : 1 Million, to 6 Million, doubling each run.
      Cutoff ratio: (Cutoff value/Size of the array): Ranging from 0.05 to to 1.0 in increments of 0.05

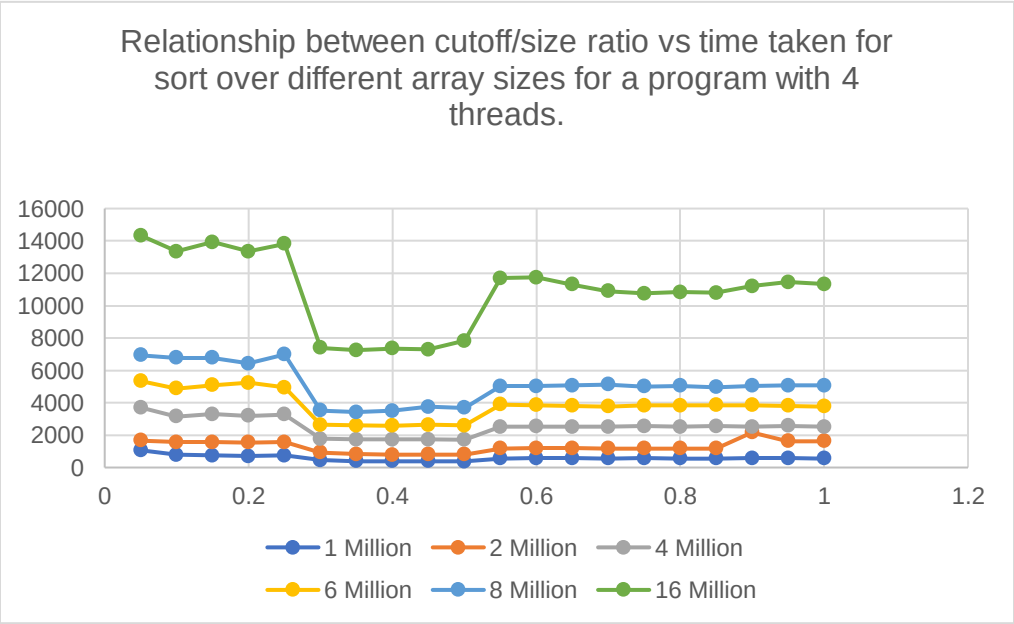
cutoff/size	1 Million	2 Million	4 Million	6 Million	8 Million	16 Million
0.05	968	1821	3134	5187	7089	13778
0.1	730	1631	3068	5073	6615	13587
0.15	789	1797	3434	5603	7266	14084
0.2	819	1747	3368	5360	7391	14998
0.25	855	1673	3159	5495	7567	15859
0.3	588	1460	2413	4078	5348	11189
0.35	616	1341	2444	4101	5265	11513
0.4	658	1329	2413	4054	5311	11146
0.45	586	1286	2426	4032	5347	11190
0.5	628	1326	2586	4051	5467	11258
0.55	577	1210	2381	3940	5130	10445
0.6	576	1282	2427	3798	5093	10442
0.65	580	1260	2402	3799	5085	10524
0.7	569	1195	2466	3876	5102	10543
0.75	565	1182	2415	3754	5082	10933
0.8	577	1251	2366	3880	5168	11425
0.85	579	1314	2371	3759	5277	11441
0.9	564	1207	2414	3823	5375	11375
0.95	561	1185	2420	3816	5556	11409
1	568	1193	2374	3895	5527	11343



- Threads : 4,
  Array sizes varied from : 1 Million, to 6 Million, doubling each run.
  Cutoff ratio: (Cutoff value/Size of the array): Ranging from 0.05 to to 1.0 in increments of 0.05

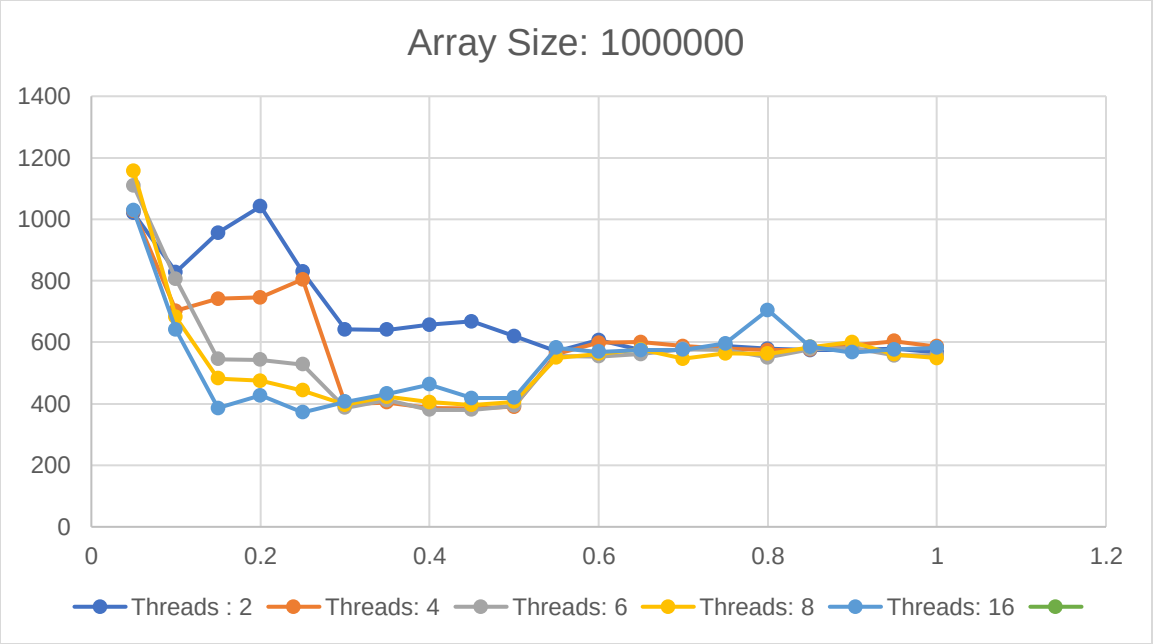
cutoff/size	1 Million	2 Million	4 Million	6 Million	8 Million	16 Million
0.05	1065	1672	3707	5341	6936	14339
0.1	778	1563	3150	4873	6781	13337
0.15	741	1564	3288	5087	6769	13932
0.2	703	1539	3188	5228	6422	13346
0.25	739	1566	3281	4935	6982	13823
0.3	446	935	1767	2638	3528	7393
0.35	399	831	1726	2605	3426	7253
0.4	393	792	1721	2576	3492	7356
0.45	395	794	1737	2640	3750	7286
0.5	381	802	1716	2595	3678	7825
0.55	560	1169	2516	3884	5018	11705
0.6	576	1194	2541	3845	5024	11743
0.65	580	1201	2508	3818	5064	11299
0.7	563	1172	2514	3778	5128	10905
0.75	571	1181	2552	3826	5006	10752
0.8	554	1177	2514	3835	5044	10838
0.85	555	1173	2547	3847	4966	10798

0.9	573	2163	2513	3857	5054	11200
0.95	571	1632	2565	3810	5070	11460
1	564	1622	2523	3776	5058	11337

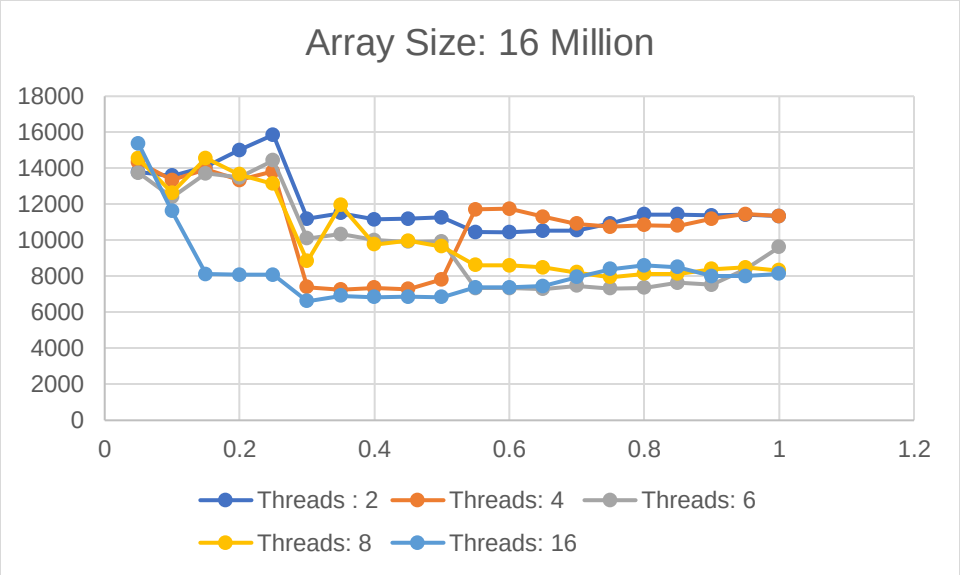


2. Checking for different threads from 2 to 16 for an array of 1000000 elements, with cutoff ratios changing from 0.05 to 1.0

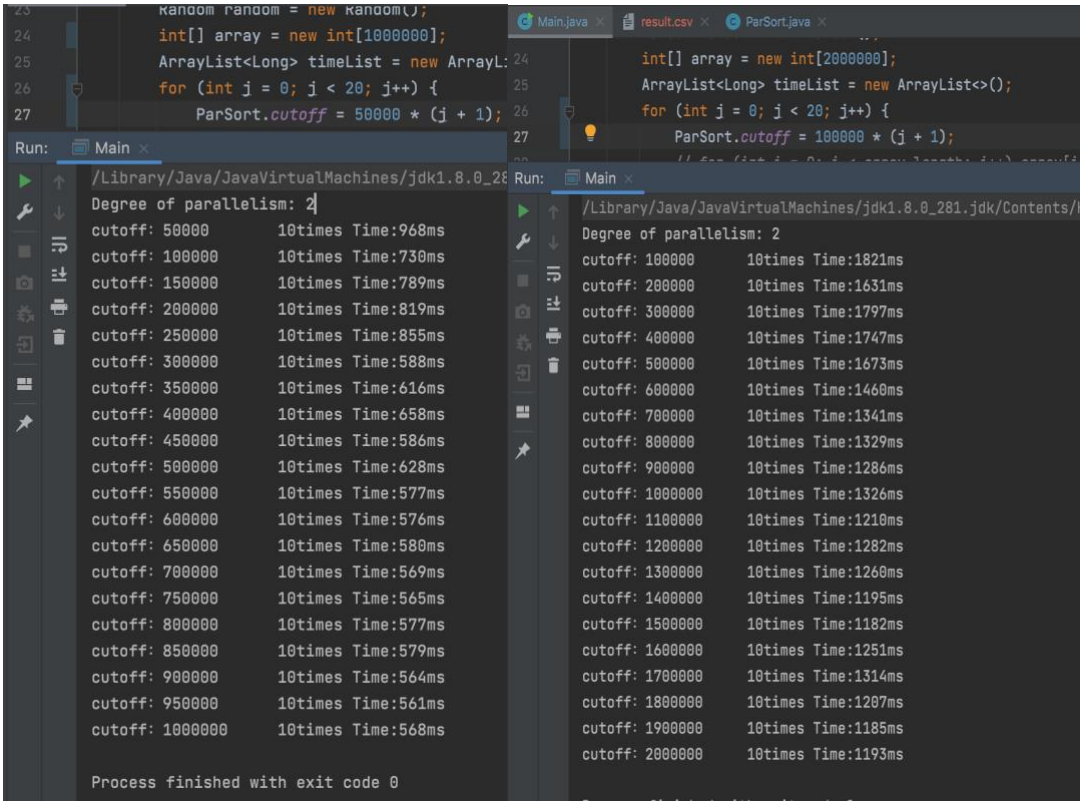
Cutoff/size ratio	Threads : 2	Threads: 4	Threads: 6	Threads: 8	Threads: 16
0.05	1020	1027	1109	1157	1029
0.1	828	702	805	682	641
0.15	956	741	545	482	386
0.2	1042	746	543	475	427
0.25	829	804	528	443	372
0.3	642	399	387	397	407
0.35	641	404	412	423	433
0.4	657	387	380	405	463
0.45	668	385	380	396	418
0.5	620	390	393	406	419
0.55	571	562	555	550	582
0.6	607	598	554	563	569
0.65	576	601	562	577	574
0.7	576	588	578	546	576
0.75	589	579	575	564	596
0.8	579	575	551	563	705
0.85	575	576	578	584	585
0.9	575	591	581	601	567
0.95	579	604	557	560	577
1	567	586	558	549	582



Cutoff/size ratio	Threads : 2	Threads: 4	Threads: 6	Threads: 8	Threads: 16
0.05	13778	14339	13742	14534	15380
0.1	13587	13337	12399	12629	11627
0.15	14084	13932	13703	14564	8116
0.2	14998	13346	13497	13648	8084
0.25	15859	13823	14431	13136	8086
0.3	11189	7393	10102	8833	6610
0.35	11513	7253	10335	11960	6916
0.4	11146	7356	10014	9769	6836
0.45	11190	7286	9907	9970	6856
0.5	11258	7825	9942	9661	6844
0.55	10445	11705	7338	8621	7380
0.6	10442	11743	7348	8596	7384
0.65	10524	11299	7287	8484	7454
0.7	10543	10905	7475	8218	7950
0.75	10933	10752	7320	7945	8392
0.8	11425	10838	7362	8133	8600
0.85	11441	10798	7639	8144	8515
0.9	11375	11200	7534	8405	8007
0.95	11409	11460	8395	8487	8004
1	11343	11337	9620	8331	8136



Output for Conclusion 1:



```
Random random = new Random();
int[] array = new int[1000000];
ArrayList<Long> timeList = new ArrayL:
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 50000 * (j + 1);
```

Run: Main

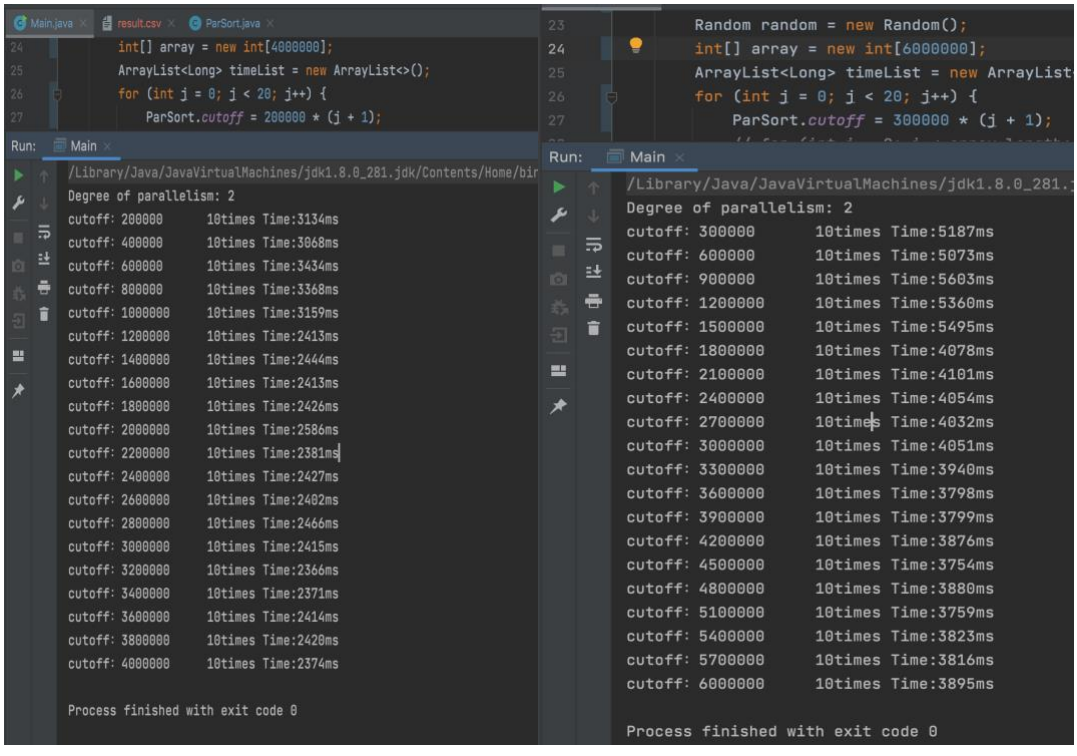
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_28
Degree of parallelism: 2
cutoff: 50000       10times Time:968ms
cutoff: 100000      10times Time:730ms
cutoff: 150000      10times Time:789ms
cutoff: 200000      10times Time:819ms
cutoff: 250000      10times Time:855ms
cutoff: 300000      10times Time:588ms
cutoff: 350000      10times Time:616ms
cutoff: 400000      10times Time:658ms
cutoff: 450000      10times Time:586ms
cutoff: 500000      10times Time:628ms
cutoff: 550000      10times Time:577ms
cutoff: 600000      10times Time:576ms
cutoff: 650000      10times Time:580ms
cutoff: 700000      10times Time:569ms
cutoff: 750000      10times Time:565ms
cutoff: 800000      10times Time:577ms
cutoff: 850000      10times Time:579ms
cutoff: 900000      10times Time:564ms
cutoff: 950000      10times Time:561ms
cutoff: 1000000     10times Time:568ms

Process finished with exit code 0
```

Main.java | result.csv | ParSort.java

```
int[] array = new int[2000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 100000 * (j + 1);
```

Run: Main

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.jdk/Contents/
Degree of parallelism: 2
cutoff: 100000      10times Time:1821ms
cutoff: 200000      10times Time:1631ms
cutoff: 300000      10times Time:1797ms
cutoff: 400000      10times Time:1747ms
cutoff: 500000      10times Time:1673ms
cutoff: 600000      10times Time:1460ms
cutoff: 700000      10times Time:1341ms
cutoff: 800000      10times Time:1329ms
cutoff: 900000      10times Time:1286ms
cutoff: 1000000     10times Time:1326ms
cutoff: 1100000     10times Time:1210ms
cutoff: 1200000     10times Time:1282ms
cutoff: 1300000     10times Time:1260ms
cutoff: 1400000     10times Time:1195ms
cutoff: 1500000     10times Time:1182ms
cutoff: 1600000     10times Time:1251ms
cutoff: 1700000     10times Time:1314ms
cutoff: 1800000     10times Time:1207ms
cutoff: 1900000     10times Time:1185ms
cutoff: 2000000     10times Time:1193ms
```



Main.java | result.csv | ParSort.java

```
int[] array = new int[4000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 200000 * (j + 1);
```

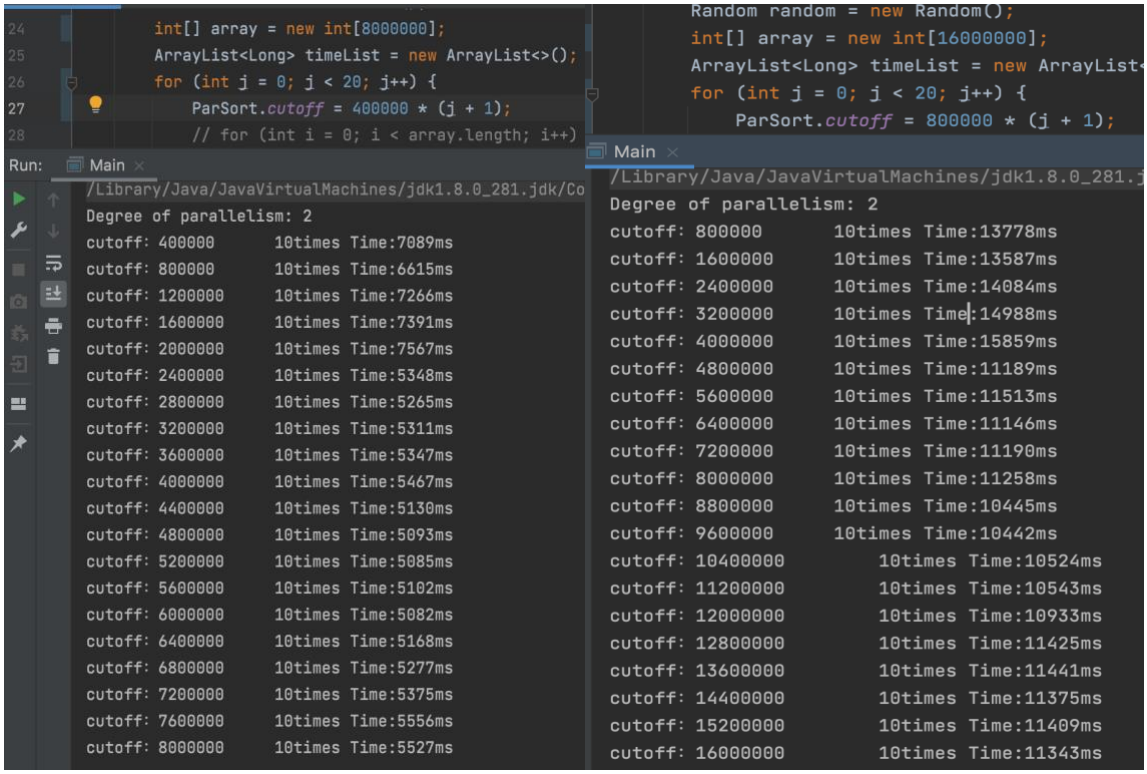
Run: Main

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.jdk/Contents/Home/bir
Degree of parallelism: 2
cutoff: 200000      10times Time:3134ms
cutoff: 400000      10times Time:3068ms
cutoff: 600000      10times Time:3434ms
cutoff: 800000      10times Time:3368ms
cutoff: 1000000     10times Time:3159ms
cutoff: 1200000     10times Time:2413ms
cutoff: 1400000     10times Time:2444ms
cutoff: 1600000     10times Time:2413ms
cutoff: 1800000     10times Time:2426ms
cutoff: 2000000     10times Time:2586ms
cutoff: 2200000     10times Time:2381ms
cutoff: 2400000     10times Time:2427ms
cutoff: 2600000     10times Time:2402ms
cutoff: 2800000     10times Time:2466ms
cutoff: 3000000     10times Time:2415ms
cutoff: 3200000     10times Time:2366ms
cutoff: 3400000     10times Time:2371ms
cutoff: 3600000     10times Time:2414ms
cutoff: 3800000     10times Time:2428ms
cutoff: 4000000     10times Time:2374ms

Process finished with exit code 0
```

```
Random random = new Random();
int[] array = new int[6000000];
ArrayList<Long> timeList = new ArrayList
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 300000 * (j + 1);
```

Run: Main

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.
Degree of parallelism: 2
cutoff: 300000      10times Time:5187ms
cutoff: 600000      10times Time:5073ms
cutoff: 900000      10times Time:5603ms
cutoff: 1200000     10times Time:5360ms
cutoff: 1500000     10times Time:5495ms
cutoff: 1800000     10times Time:4078ms
cutoff: 2100000     10times Time:4101ms
cutoff: 2400000     10times Time:4054ms
cutoff: 2700000     10times Time:4032ms
cutoff: 3000000     10times Time:4051ms
cutoff: 3300000     10times Time:3940ms
cutoff: 3600000     10times Time:3798ms
cutoff: 3900000     10times Time:3799ms
cutoff: 4200000     10times Time:3876ms
cutoff: 4500000     10times Time:3754ms
cutoff: 4800000     10times Time:3880ms
cutoff: 5100000     10times Time:3759ms
cutoff: 5400000     10times Time:3823ms
cutoff: 5700000     10times Time:3816ms
cutoff: 6000000     10times Time:3895ms

Process finished with exit code 0
```



```
int[] array = new int[8000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 400000 * (j + 1);
    // for (int i = 0; i < array.length; i++)
```

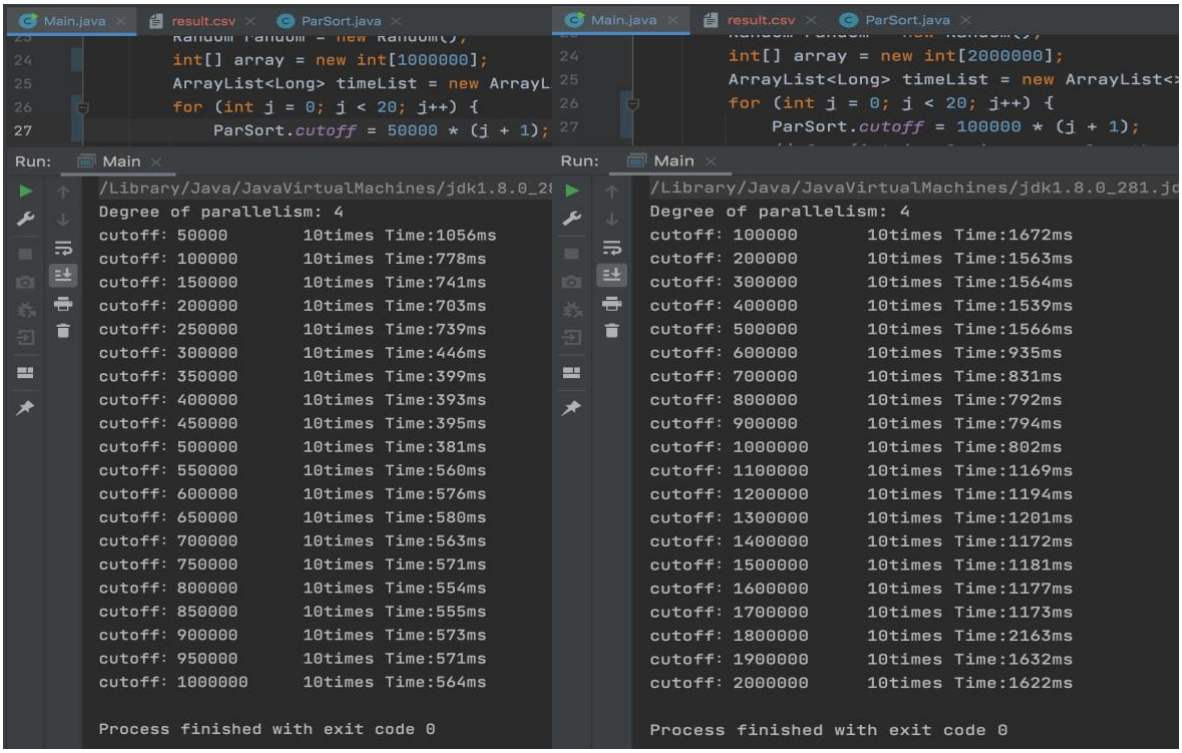
Run: Main

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.jdk/Co
Degree of parallelism: 2
cutoff: 400000      10times Time:7089ms
cutoff: 800000      10times Time:6615ms
cutoff: 1200000     10times Time:7266ms
cutoff: 1600000     10times Time:7391ms
cutoff: 2000000     10times Time:7567ms
cutoff: 2400000     10times Time:5348ms
cutoff: 2800000     10times Time:5265ms
cutoff: 3200000     10times Time:5311ms
cutoff: 3600000     10times Time:5347ms
cutoff: 4000000     10times Time:5467ms
cutoff: 4400000     10times Time:5130ms
cutoff: 4800000     10times Time:5093ms
cutoff: 5200000     10times Time:5085ms
cutoff: 5600000     10times Time:5102ms
cutoff: 6000000     10times Time:5082ms
cutoff: 6400000     10times Time:5168ms
cutoff: 6800000     10times Time:5277ms
cutoff: 7200000     10times Time:5375ms
cutoff: 7600000     10times Time:5556ms
cutoff: 8000000     10times Time:5527ms
```

```
Random random = new Random();
int[] array = new int[16000000];
ArrayList<Long> timeList = new ArrayList<
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 800000 * (j + 1);
```

Main

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.j
Degree of parallelism: 2
cutoff: 800000      10times Time:13778ms
cutoff: 1600000     10times Time:13587ms
cutoff: 2400000     10times Time:14084ms
cutoff: 3200000     10times Time:14988ms
cutoff: 4000000     10times Time:15859ms
cutoff: 4800000     10times Time:11189ms
cutoff: 5600000     10times Time:11513ms
cutoff: 6400000     10times Time:11146ms
cutoff: 7200000     10times Time:11190ms
cutoff: 8000000     10times Time:11258ms
cutoff: 8800000     10times Time:10445ms
cutoff: 9600000     10times Time:10442ms
cutoff: 10400000        10times Time:10524ms
cutoff: 11200000        10times Time:10543ms
cutoff: 12000000        10times Time:10933ms
cutoff: 12800000        10times Time:11425ms
cutoff: 13600000        10times Time:11441ms
cutoff: 14400000        10times Time:11375ms
cutoff: 15200000        10times Time:11409ms
cutoff: 16000000        10times Time:11343ms
```



Main.java | result.csv | ParSort.java

```
int[] array = new int[1000000];
ArrayList<Long> timeList = new ArrayL
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 50000 * (j + 1);
```

Run: Main

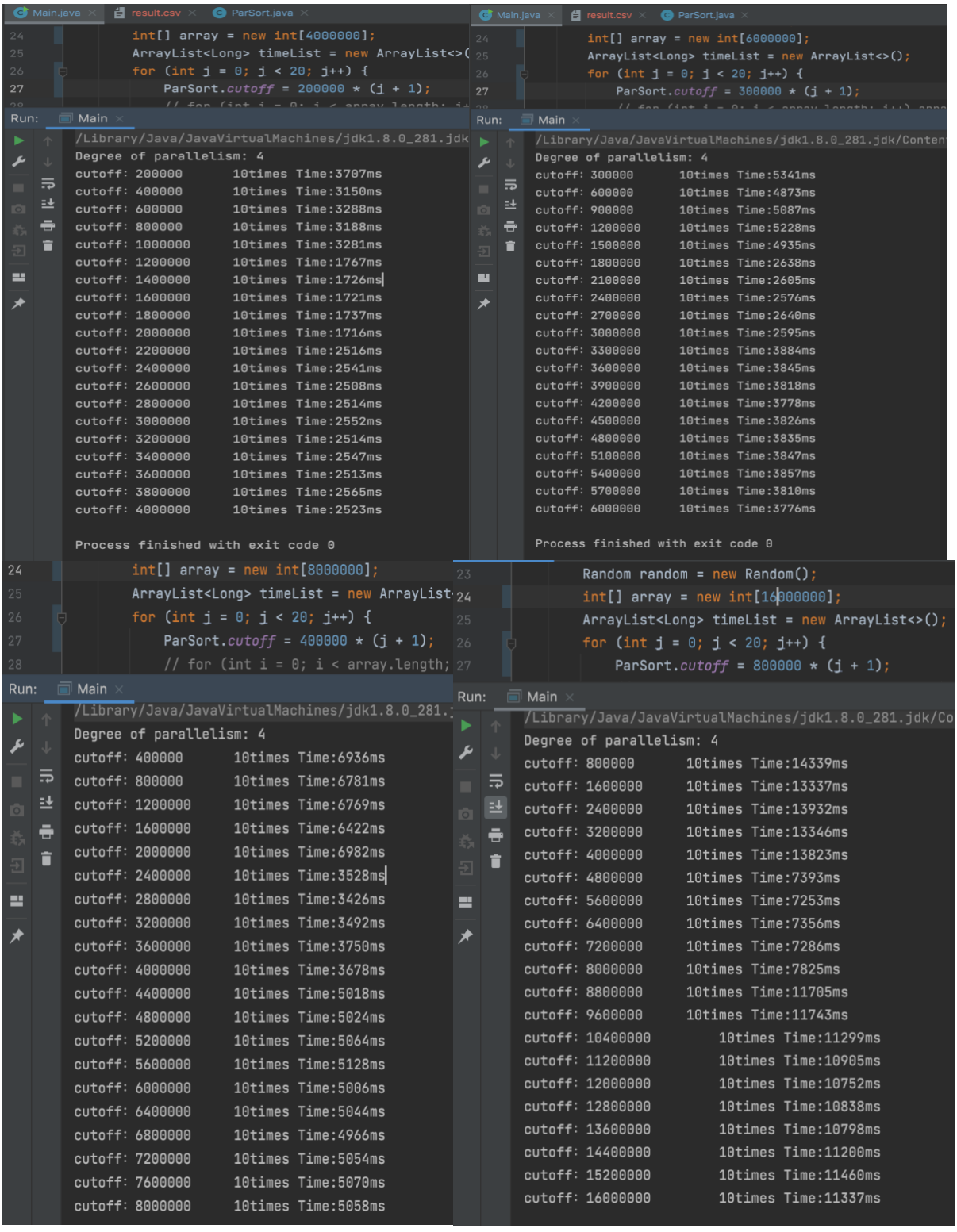
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_28
Degree of parallelism: 4
cutoff: 50000       10times Time:1056ms
cutoff: 100000      10times Time:778ms
cutoff: 150000      10times Time:741ms
cutoff: 200000      10times Time:703ms
cutoff: 250000      10times Time:739ms
cutoff: 300000      10times Time:446ms
cutoff: 350000      10times Time:399ms
cutoff: 400000      10times Time:393ms
cutoff: 450000      10times Time:395ms
cutoff: 500000      10times Time:381ms
cutoff: 550000      10times Time:560ms
cutoff: 600000      10times Time:576ms
cutoff: 650000      10times Time:580ms
cutoff: 700000      10times Time:563ms
cutoff: 750000      10times Time:571ms
cutoff: 800000      10times Time:554ms
cutoff: 850000      10times Time:555ms
cutoff: 900000      10times Time:573ms
cutoff: 950000      10times Time:571ms
cutoff: 1000000     10times Time:564ms

Process finished with exit code 0
```

Main.java | result.csv | ParSort.java

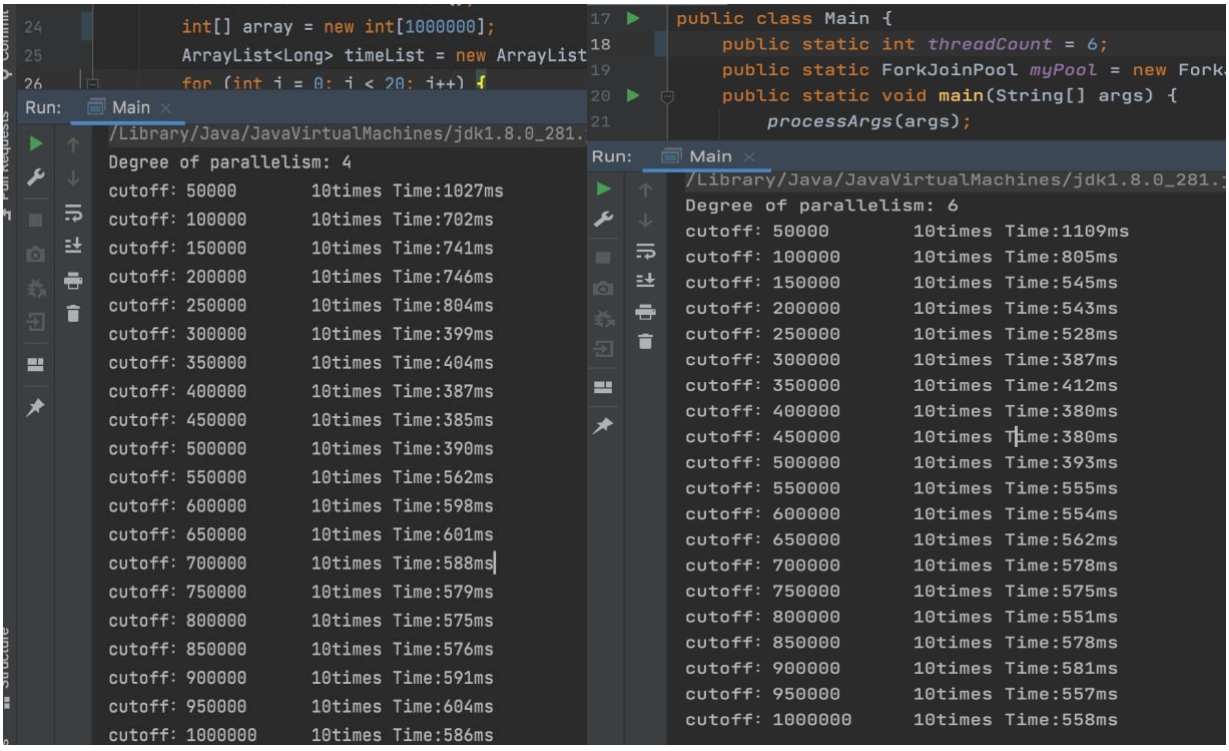
```
int[] array = new int[2000000];
ArrayList<Long> timeList = new ArrayList<
for (int j = 0; j < 20; j++) {
    ParSort.cutoff = 100000 * (j + 1);
```

Run: Main

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.jd
Degree of parallelism: 4
cutoff: 100000      10times Time:1672ms
cutoff: 200000      10times Time:1563ms
cutoff: 300000      10times Time:1564ms
cutoff: 400000      10times Time:1539ms
cutoff: 500000      10times Time:1566ms
cutoff: 600000      10times Time:935ms
cutoff: 700000      10times Time:831ms
cutoff: 800000      10times Time:792ms
cutoff: 900000      10times Time:794ms
cutoff: 1000000     10times Time:802ms
cutoff: 1100000     10times Time:1169ms
cutoff: 1200000     10times Time:1194ms
cutoff: 1300000     10times Time:1201ms
cutoff: 1400000     10times Time:1172ms
cutoff: 1500000     10times Time:1181ms
cutoff: 1600000     10times Time:1177ms
cutoff: 1700000     10times Time:1173ms
cutoff: 1800000     10times Time:2163ms
cutoff: 1900000     10times Time:1632ms
cutoff: 2000000     10times Time:1622ms

Process finished with exit code 0
```


Main.java
result.csv
ParSort.java
int[] array = new int[4000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 20; j++) {
ParSort.cutoff = 200000 * (j + 1);
Run: Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.jdk
Degree of parallelism: 4
cutoff: 200000 10times Time:3707ms
cutoff: 400000 10times Time:3150ms
cutoff: 600000 10times Time:3288ms
cutoff: 800000 10times Time:3188ms
cutoff: 1000000 10times Time:3281ms
cutoff: 1200000 10times Time:1767ms
cutoff: 1400000 10times Time:1726ms
cutoff: 1600000 10times Time:1721ms
cutoff: 1800000 10times Time:1737ms
cutoff: 2000000 10times Time:1716ms
cutoff: 2200000 10times Time:2516ms
cutoff: 2400000 10times Time:2541ms
cutoff: 2600000 10times Time:2508ms
cutoff: 2800000 10times Time:2514ms
cutoff: 3000000 10times Time:2552ms
cutoff: 3200000 10times Time:2514ms
cutoff: 3400000 10times Time:2547ms
cutoff: 3600000 10times Time:2513ms
cutoff: 3800000 10times Time:2565ms
cutoff: 4000000 10times Time:2523ms
Process finished with exit code 0
int[] array = new int[6000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 20; j++) {
ParSort.cutoff = 300000 * (j + 1);
Run: Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.jdk/Conten
Degree of parallelism: 4
cutoff: 300000 10times Time:5341ms
cutoff: 600000 10times Time:4873ms
cutoff: 900000 10times Time:5087ms
cutoff: 1200000 10times Time:5228ms
cutoff: 1500000 10times Time:4935ms
cutoff: 1800000 10times Time:2638ms
cutoff: 2100000 10times Time:2605ms
cutoff: 2400000 10times Time:2576ms
cutoff: 2700000 10times Time:2640ms
cutoff: 3000000 10times Time:2595ms
cutoff: 3300000 10times Time:3884ms
cutoff: 3600000 10times Time:3845ms
cutoff: 3900000 10times Time:3818ms
cutoff: 4200000 10times Time:3778ms
cutoff: 4500000 10times Time:3826ms
cutoff: 4800000 10times Time:3835ms
cutoff: 5100000 10times Time:3847ms
cutoff: 5400000 10times Time:3857ms
cutoff: 5700000 10times Time:3810ms
cutoff: 6000000 10times Time:3776ms
Process finished with exit code 0
int[] array = new int[8000000];
ArrayList<Long> timeList = new ArrayList
for (int j = 0; j < 20; j++) {
ParSort.cutoff = 400000 * (j + 1);
// for (int i = 0; i < array.length;
Run: Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.
Degree of parallelism: 4
cutoff: 400000 10times Time:6936ms
cutoff: 800000 10times Time:6781ms
cutoff: 1200000 10times Time:6769ms
cutoff: 1600000 10times Time:6422ms
cutoff: 2000000 10times Time:6982ms
cutoff: 2400000 10times Time:3528ms
cutoff: 2800000 10times Time:3426ms
cutoff: 3200000 10times Time:3492ms
cutoff: 3600000 10times Time:3750ms
cutoff: 4000000 10times Time:3678ms
cutoff: 4400000 10times Time:5018ms
cutoff: 4800000 10times Time:5024ms
cutoff: 5200000 10times Time:5064ms
cutoff: 5600000 10times Time:5128ms
cutoff: 6000000 10times Time:5006ms
cutoff: 6400000 10times Time:5044ms
cutoff: 6800000 10times Time:4966ms
cutoff: 7200000 10times Time:5054ms
cutoff: 7600000 10times Time:5070ms
cutoff: 8000000 10times Time:5058ms
Random random = new Random();
int[] array = new int[16000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 20; j++) {
ParSort.cutoff = 800000 * (j + 1);
Run: Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.jdk/Co
Degree of parallelism: 4
cutoff: 800000 10times Time:14339ms
cutoff: 1600000 10times Time:13337ms
cutoff: 2400000 10times Time:13932ms
cutoff: 3200000 10times Time:13346ms
cutoff: 4000000 10times Time:13823ms
cutoff: 4800000 10times Time:7393ms
cutoff: 5600000 10times Time:7253ms
cutoff: 6400000 10times Time:7356ms
cutoff: 7200000 10times Time:7286ms
cutoff: 8000000 10times Time:7825ms
cutoff: 8800000 10times Time:11705ms
cutoff: 9600000 10times Time:11743ms
cutoff: 10400000 10times Time:11299ms
cutoff: 11200000 10times Time:10905ms
cutoff: 12000000 10times Time:10752ms
cutoff: 12800000 10times Time:10838ms
cutoff: 13600000 10times Time:10798ms
cutoff: 14400000 10times Time:11200ms
cutoff: 15200000 10times Time:11460ms
cutoff: 16000000 10times Time:11337ms

Sample Output for Conclusion 2:
For array size: 1 Million, varying threads from 2 to 16

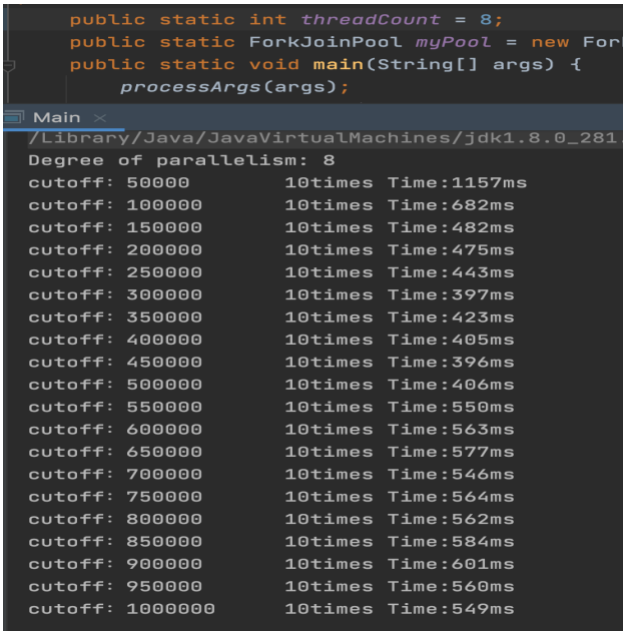


```
int[] array = new int[1000000];
ArrayList<Long> timeList = new ArrayList
for (int i = 0; i < 20; i++) {
```

```
Run:  Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.
Degree of parallelism: 4
cutoff: 50000       10times Time:1027ms
cutoff: 100000      10times Time:702ms
cutoff: 150000      10times Time:741ms
cutoff: 200000      10times Time:746ms
cutoff: 250000      10times Time:804ms
cutoff: 300000      10times Time:399ms
cutoff: 350000      10times Time:404ms
cutoff: 400000      10times Time:387ms
cutoff: 450000      10times Time:385ms
cutoff: 500000      10times Time:390ms
cutoff: 550000      10times Time:562ms
cutoff: 600000      10times Time:598ms
cutoff: 650000      10times Time:601ms
cutoff: 700000      10times Time:588ms
cutoff: 750000      10times Time:579ms
cutoff: 800000      10times Time:575ms
cutoff: 850000      10times Time:576ms
cutoff: 900000      10times Time:591ms
cutoff: 950000      10times Time:604ms
cutoff: 1000000     10times Time:586ms
```

```
public class Main {
    public static int threadCount = 6;
    public static ForkJoinPool myPool = new Fork
    public static void main(String[] args) {
        processArgs(args);
```

```
Run:  Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_281.
Degree of parallelism: 6
cutoff: 50000       10times Time:1109ms
cutoff: 100000      10times Time:805ms
cutoff: 150000      10times Time:545ms
cutoff: 200000      10times Time:543ms
cutoff: 250000      10times Time:528ms
cutoff: 300000      10times Time:387ms
cutoff: 350000      10times Time:412ms
cutoff: 400000      10times Time:380ms
cutoff: 450000      10times Time:380ms
cutoff: 500000      10times Time:393ms
cutoff: 550000      10times Time:555ms
cutoff: 600000      10times Time:554ms
cutoff: 650000      10times Time:562ms
cutoff: 700000      10times Time:578ms
cutoff: 750000      10times Time:575ms
cutoff: 800000      10times Time:551ms
cutoff: 850000      10times Time:578ms
cutoff: 900000      10times Time:581ms
cutoff: 950000      10times Time:557ms
cutoff: 1000000     10times Time:558ms
```



```
public static int threadCount = 8;
public static ForkJoinPool myPool = new For
public static void main(String[] args) {
    processArgs(args);
```

```
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_281
Degree of parallelism: 8
cutoff: 50000       10times Time:1157ms
cutoff: 100000      10times Time:682ms
cutoff: 150000      10times Time:482ms
cutoff: 200000      10times Time:475ms
cutoff: 250000      10times Time:443ms
cutoff: 300000      10times Time:397ms
cutoff: 350000      10times Time:423ms
cutoff: 400000      10times Time:405ms
cutoff: 450000      10times Time:396ms
cutoff: 500000      10times Time:406ms
cutoff: 550000      10times Time:550ms
cutoff: 600000      10times Time:563ms
cutoff: 650000      10times Time:577ms
cutoff: 700000      10times Time:546ms
cutoff: 750000      10times Time:564ms
cutoff: 800000      10times Time:562ms
cutoff: 850000      10times Time:584ms
cutoff: 900000      10times Time:601ms
cutoff: 950000      10times Time:560ms
cutoff: 1000000     10times Time:549ms
```

For Array size: 16 Million, varying threads from 2 to 16: