Deep Shah (002766755)

Program Structure & Algorithms

Spring 2023(Sec 03)

Assignment-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.

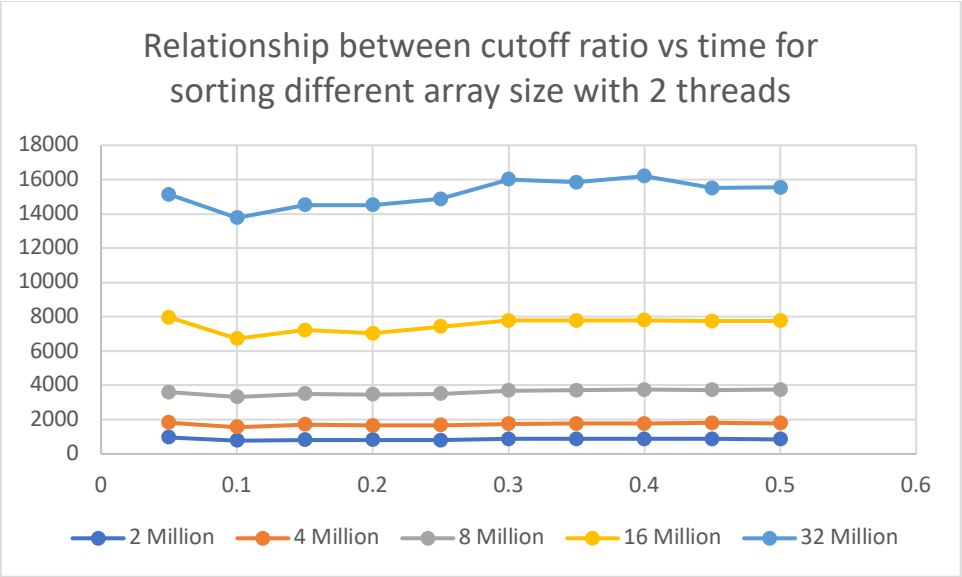
Relationship Conclusion:

1. According to the cutoff ratio, we can see that there is a range of cutoff ratios between 0.3 and 0.5 for which parallel sorting can be the best option, independent of array size and the number of threads available for parallel sorting.
2. For a number of threads ranging from 2 to 32, examination of various array sizes (2 million to 32 million) reveals that 8 threads perform best.

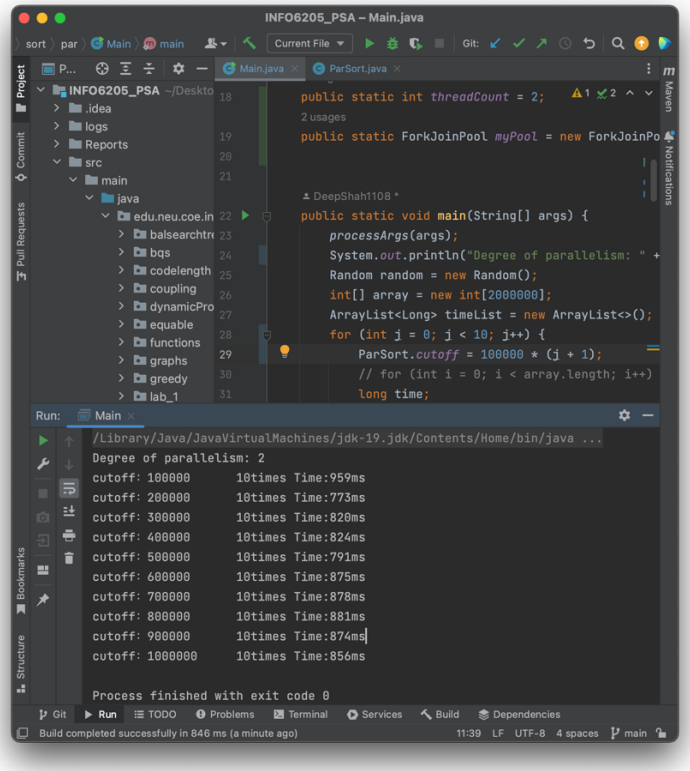
Evidence to Support that conclusion:

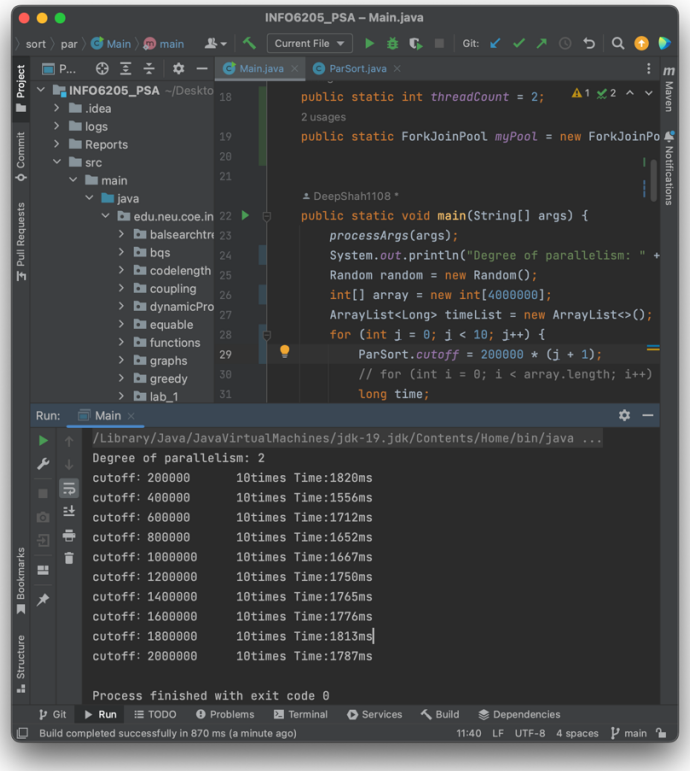
For thread 2, Array Size from 2million to 32million for cutoff ratio from 0.05 to 0.5

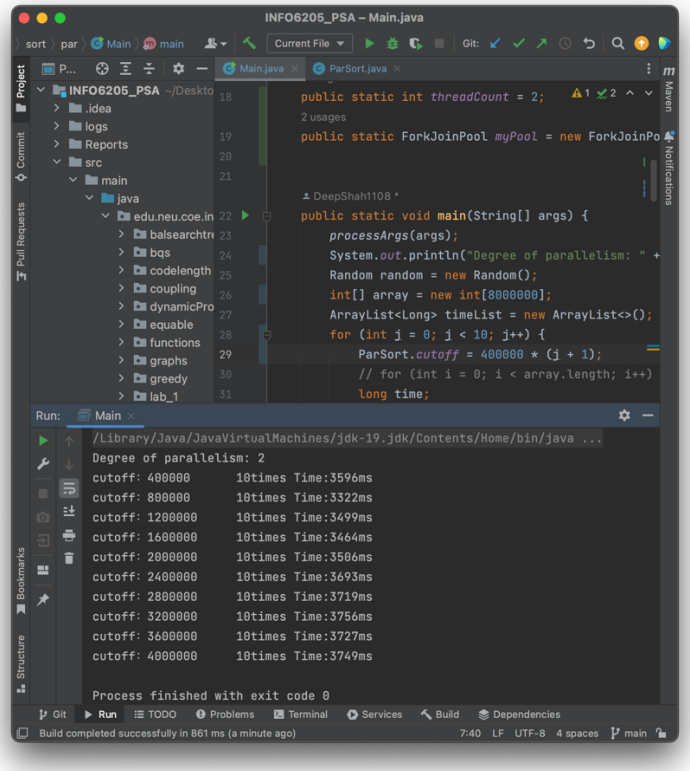
Cutoff Ratio	2 million	4 million	8 million	16 million	32 million
0.05	959	1820	3596	7974	15138
0.1	773	1556	3322	6719	13770
0.15	820	1712	3499	7216	14510
0.2	824	1652	3464	7031	14512
0.25	791	1667	3506	7420	14876
0.3	875	1750	3693	7777	16006
0.35	878	1765	3719	7787	15831
0.4	881	1776	3756	7800	16198
0.45	874	1813	3727	7736	15510
0.5	856	1787	3749	7751	15535

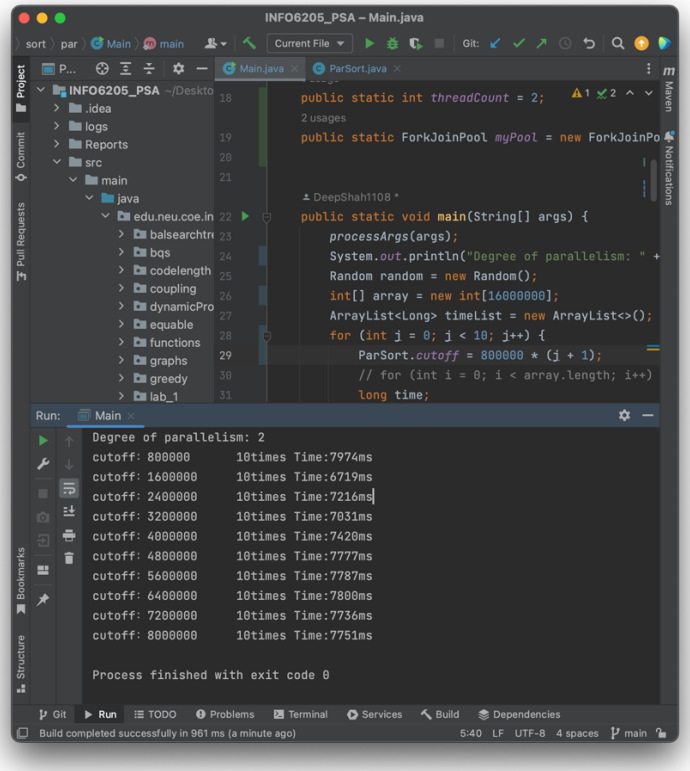

Relationship between cutoff ratio vs time for sorting different array size with 2 threads
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
0.1
0.2
0.3
0.4
0.5
0.6
2 Million
4 Million
8 Million
16 Million
32 Million

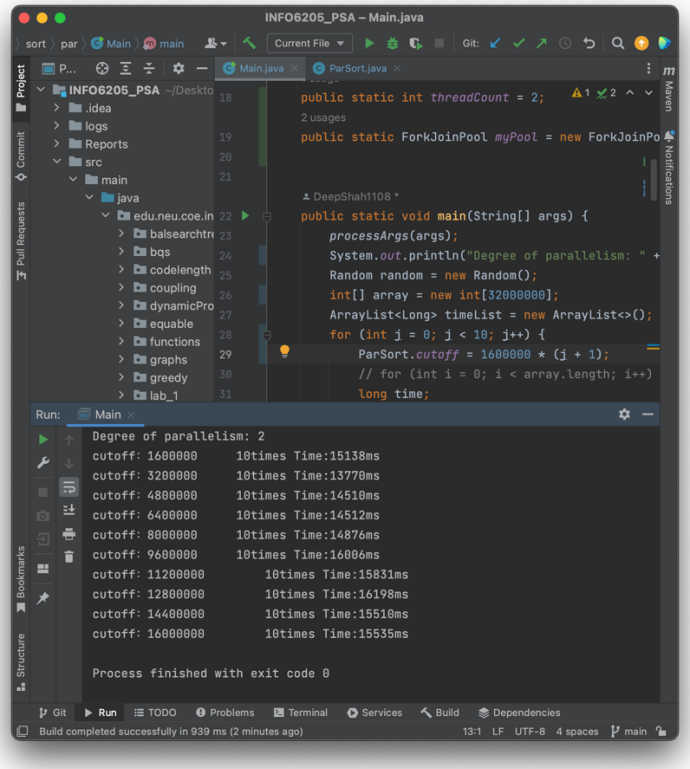
Output:


INFO6205_PSA – Main.java
public static int threadCount = 2;
public static ForkJoinPool myPool = new ForkJoinPo
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[2000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 100000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 2
cutoff: 100000 10times Time:959ms
cutoff: 200000 10times Time:773ms
cutoff: 300000 10times Time:820ms
cutoff: 400000 10times Time:824ms
cutoff: 500000 10times Time:791ms
cutoff: 600000 10times Time:875ms
cutoff: 700000 10times Time:878ms
cutoff: 800000 10times Time:881ms
cutoff: 900000 10times Time:874ms
cutoff: 1000000 10times Time:856ms
Process finished with exit code 0


INFO6205_PSA – Main.java
public static int threadCount = 2;
public static ForkJoinPool myPool = new ForkJoinPo
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[4000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 200000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 2
cutoff: 200000 10times Time:1820ms
cutoff: 400000 10times Time:1556ms
cutoff: 600000 10times Time:1712ms
cutoff: 800000 10times Time:1652ms
cutoff: 1000000 10times Time:1667ms
cutoff: 1200000 10times Time:1750ms
cutoff: 1400000 10times Time:1765ms
cutoff: 1600000 10times Time:1776ms
cutoff: 1800000 10times Time:1813ms
cutoff: 2000000 10times Time:1787ms
Process finished with exit code 0


INFO6205_PSA – Main.java
Main.java
ParSort.java
public static int threadCount = 2;
2 usages
public static ForkJoinPool myPool = new ForkJoinPo
DeepShah1108 *
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[8000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 400000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 2
cutoff: 400000 10times Time:3596ms
cutoff: 800000 10times Time:3322ms
cutoff: 1200000 10times Time:3499ms
cutoff: 1600000 10times Time:3464ms
cutoff: 2000000 10times Time:3506ms
cutoff: 2400000 10times Time:3693ms
cutoff: 2800000 10times Time:3719ms
cutoff: 3200000 10times Time:3756ms
cutoff: 3600000 10times Time:3727ms
cutoff: 4000000 10times Time:3749ms
Process finished with exit code 0
Build completed successfully in 861 ms (a minute ago)

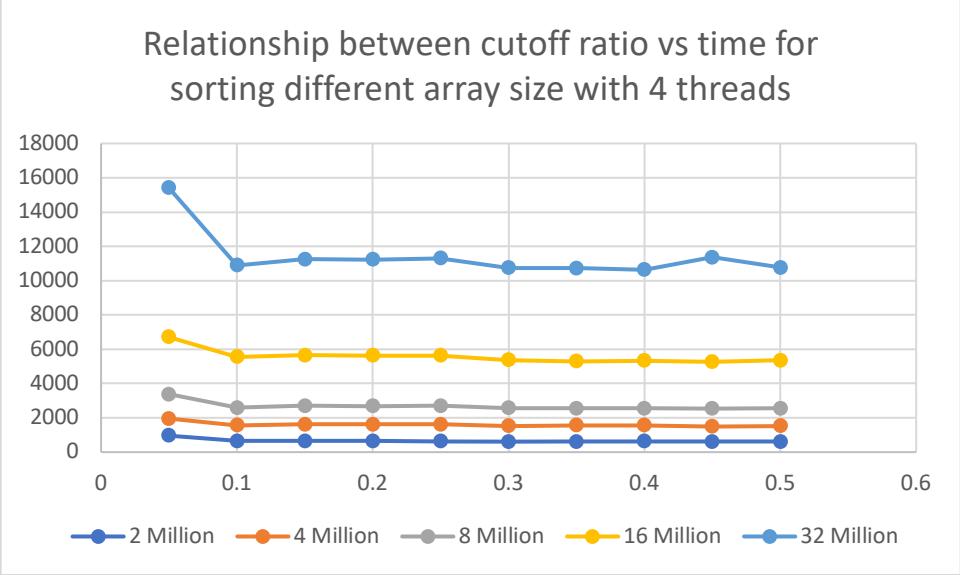

INFO6205_PSA – Main.java
Main.java
ParSort.java
public static int threadCount = 2;
2 usages
public static ForkJoinPool myPool = new ForkJoinPo
DeepShah1108 *
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[16000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 800000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
Degree of parallelism: 2
cutoff: 800000 10times Time:7974ms
cutoff: 1600000 10times Time:6719ms
cutoff: 2400000 10times Time:7216ms
cutoff: 3200000 10times Time:7031ms
cutoff: 4000000 10times Time:7420ms
cutoff: 4800000 10times Time:7777ms
cutoff: 5600000 10times Time:7787ms
cutoff: 6400000 10times Time:7800ms
cutoff: 7200000 10times Time:7736ms
cutoff: 8000000 10times Time:7751ms
Process finished with exit code 0
Build completed successfully in 961 ms (a minute ago)


INFO6205_PSA – Main.java
Main.java
ParSort.java
public static int threadCount = 2;
2 usages
public static ForkJoinPool myPool = new ForkJoinPo
DeepShah1108 *
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[32000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 1600000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
Degree of parallelism: 2
cutoff: 1600000 10times Time:15138ms
cutoff: 3200000 10times Time:13770ms
cutoff: 4800000 10times Time:14510ms
cutoff: 6400000 10times Time:14512ms
cutoff: 8000000 10times Time:14876ms
cutoff: 9600000 10times Time:16006ms
cutoff: 11200000 10times Time:15831ms
cutoff: 12800000 10times Time:16198ms
cutoff: 14400000 10times Time:15510ms
cutoff: 16000000 10times Time:15535ms
Process finished with exit code 0
Build completed successfully in 939 ms (2 minutes ago)

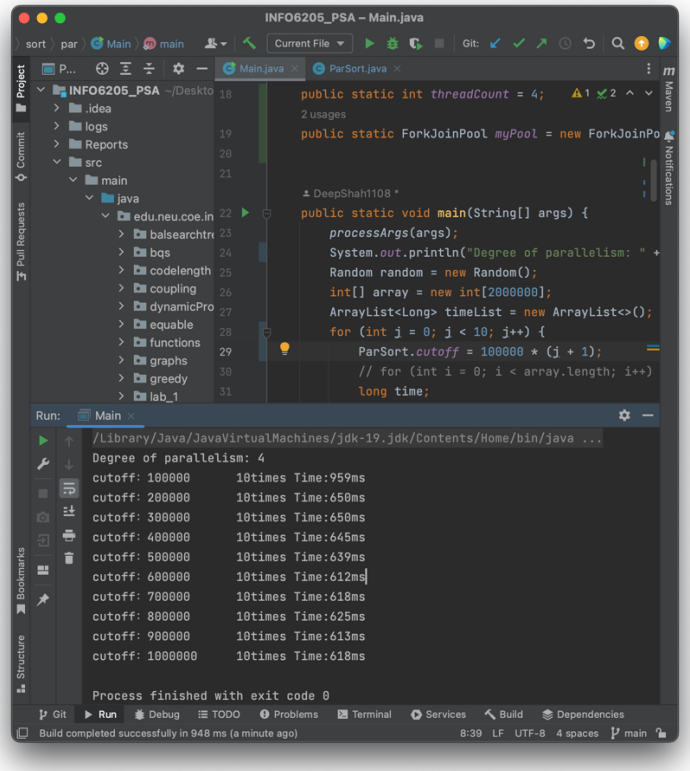
Evidence to Support that conclusion:

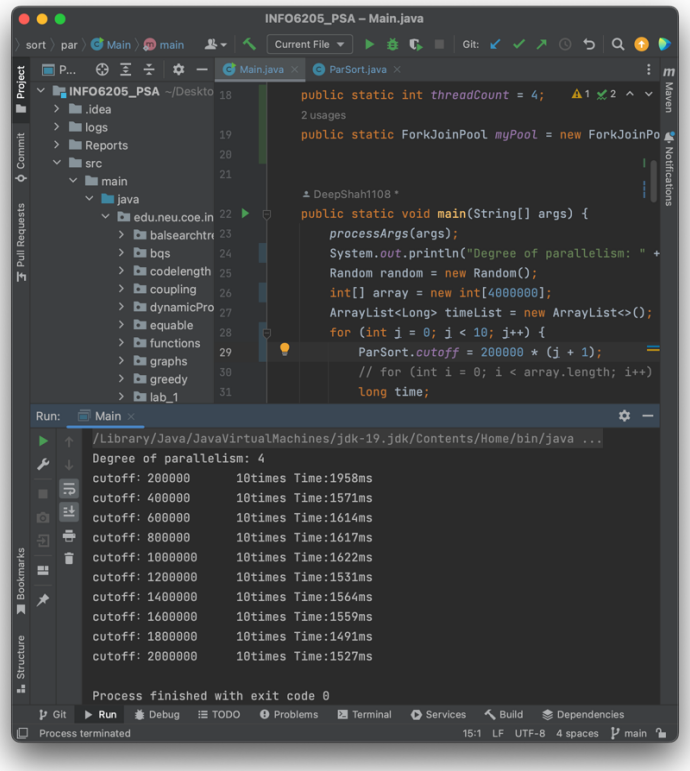
For thread 4, Array Size from 2million to 32million for cutoff ratio from 0.05 to 0.5

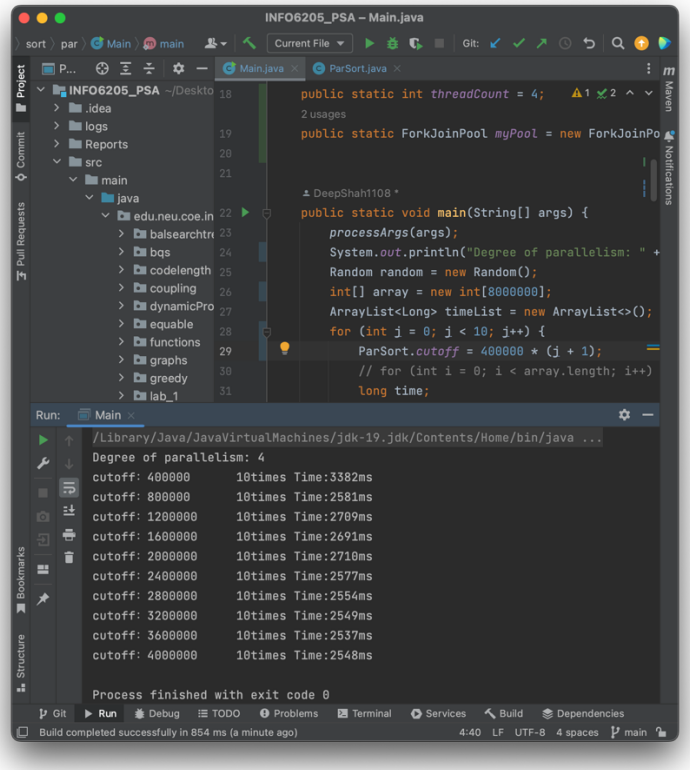
Cutoff Ratio	2 million	4 million	8 million	16 million	32 million
0.05	959	1958	3382	6705	15402
0.1	650	1571	2581	5553	10891
0.15	650	1614	2709	5654	11257
0.2	645	1617	2691	5633	11234
0.25	639	1622	2710	5625	11297
0.3	612	1531	2577	5373	10751
0.35	618	1564	2554	5295	10734
0.4	625	1559	2549	5339	10632
0.45	613	1491	2537	5265	11370
0.5	618	1527	2548	5360	10759

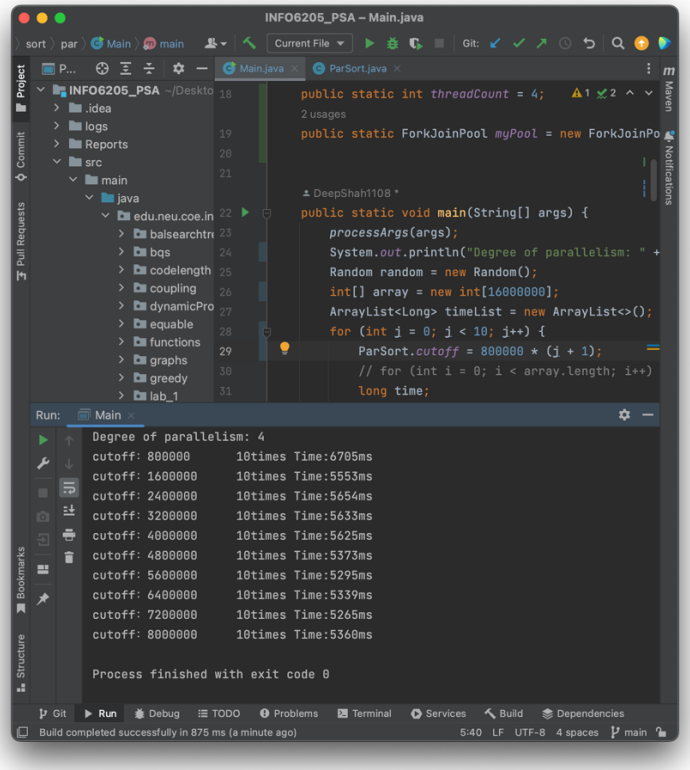


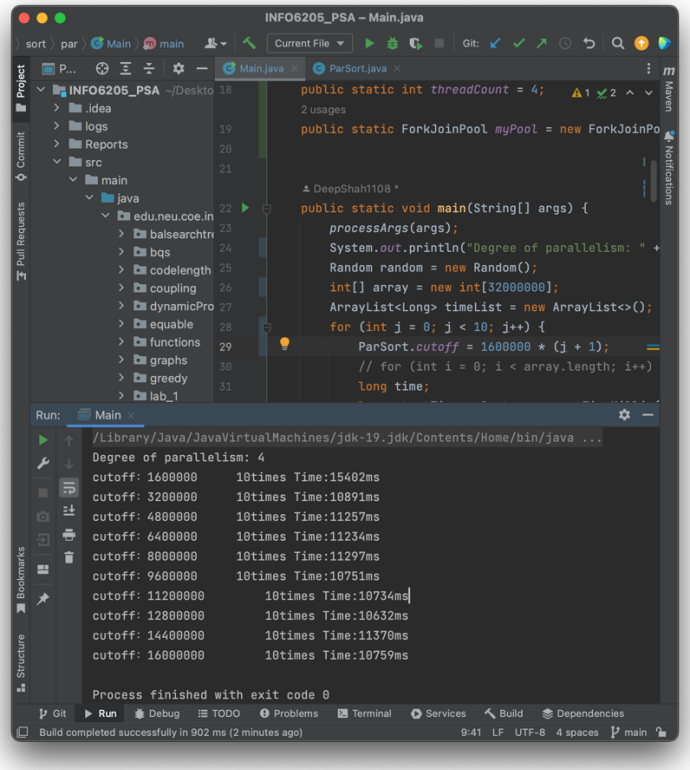
Output:


INFO6205_PSA – Main.java
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 4
cutoff: 100000 10times Time:959ms
cutoff: 200000 10times Time:650ms
cutoff: 300000 10times Time:650ms
cutoff: 400000 10times Time:645ms
cutoff: 500000 10times Time:639ms
cutoff: 600000 10times Time:612ms
cutoff: 700000 10times Time:618ms
cutoff: 800000 10times Time:625ms
cutoff: 900000 10times Time:613ms
cutoff: 1000000 10times Time:618ms
Process finished with exit code 0
Build completed successfully in 948 ms (a minute ago)


INFO6205_PSA – Main.java
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 4
cutoff: 200000 10times Time:1958ms
cutoff: 400000 10times Time:1571ms
cutoff: 600000 10times Time:1614ms
cutoff: 800000 10times Time:1617ms
cutoff: 1000000 10times Time:1622ms
cutoff: 1200000 10times Time:1531ms
cutoff: 1400000 10times Time:1564ms
cutoff: 1600000 10times Time:1559ms
cutoff: 1800000 10times Time:1491ms
cutoff: 2000000 10times Time:1527ms
Process finished with exit code 0
Process terminated


INFO6205_PSA – Main.java
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 4
cutoff: 400000 10times Time:3382ms
cutoff: 800000 10times Time:2581ms
cutoff: 1200000 10times Time:2709ms
cutoff: 1600000 10times Time:2691ms
cutoff: 2000000 10times Time:2710ms
cutoff: 2400000 10times Time:2577ms
cutoff: 2800000 10times Time:2554ms
cutoff: 3200000 10times Time:2549ms
cutoff: 3600000 10times Time:2537ms
cutoff: 4000000 10times Time:2548ms
Process finished with exit code 0
Build completed successfully in 854 ms (a minute ago)

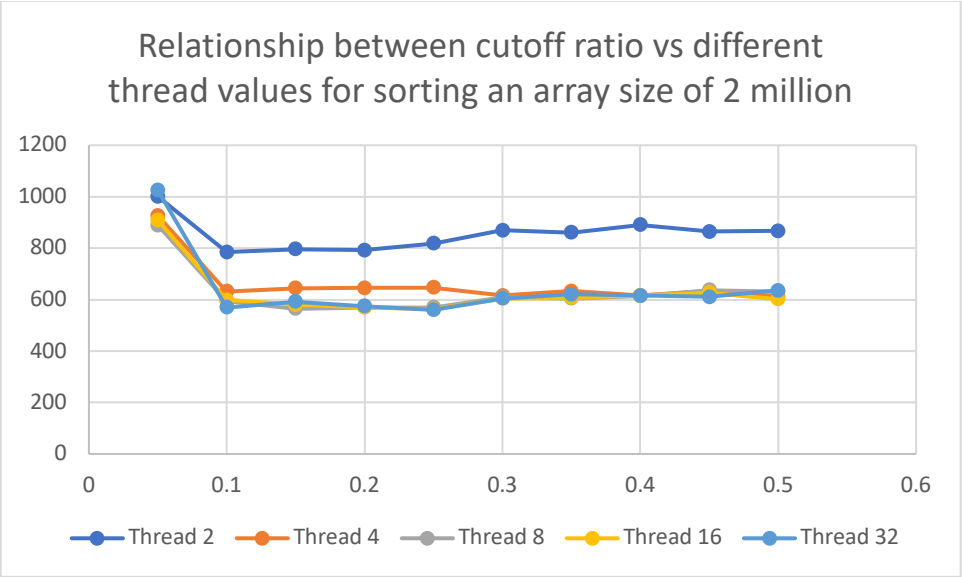

INFO6205_PSA – Main.java
Run: Main
Degree of parallelism: 4
cutoff: 800000 10times Time:6705ms
cutoff: 1600000 10times Time:5553ms
cutoff: 2400000 10times Time:5654ms
cutoff: 3200000 10times Time:5633ms
cutoff: 4000000 10times Time:5625ms
cutoff: 4800000 10times Time:5373ms
cutoff: 5600000 10times Time:5295ms
cutoff: 6400000 10times Time:5339ms
cutoff: 7200000 10times Time:5265ms
cutoff: 8000000 10times Time:5360ms
Process finished with exit code 0
Build completed successfully in 875 ms (a minute ago)



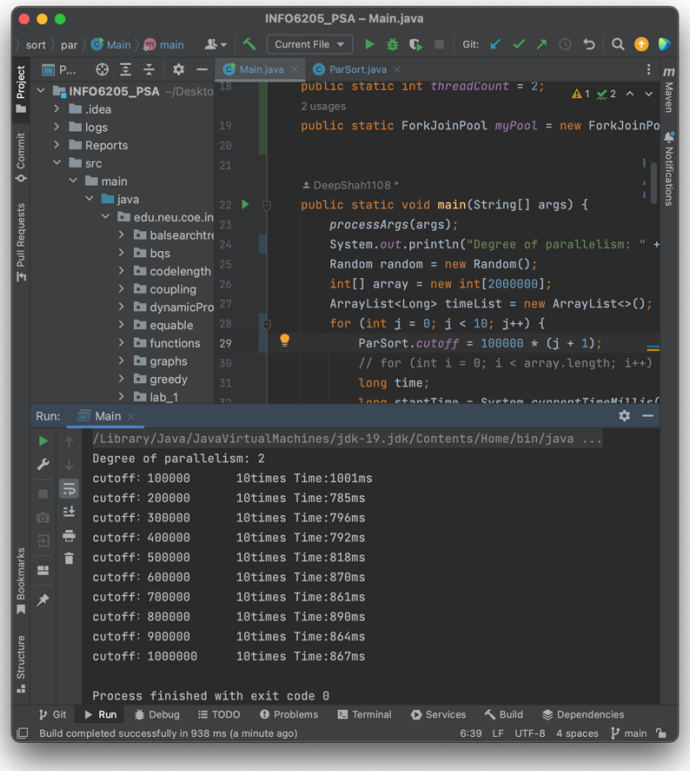
Evidence to Support that conclusion:

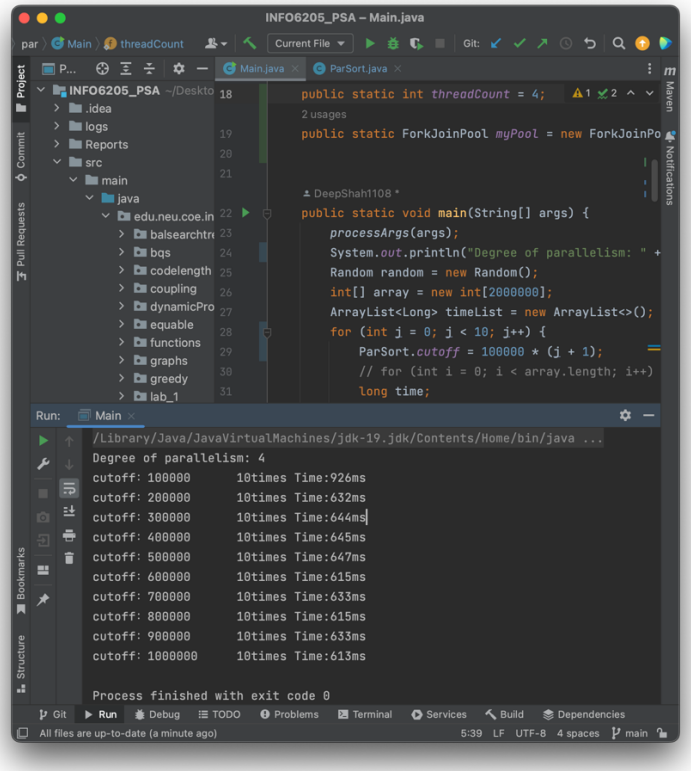
Varying thread from 2 to 32, for array size 2million with cutoff ration from 0.05 to 0.5

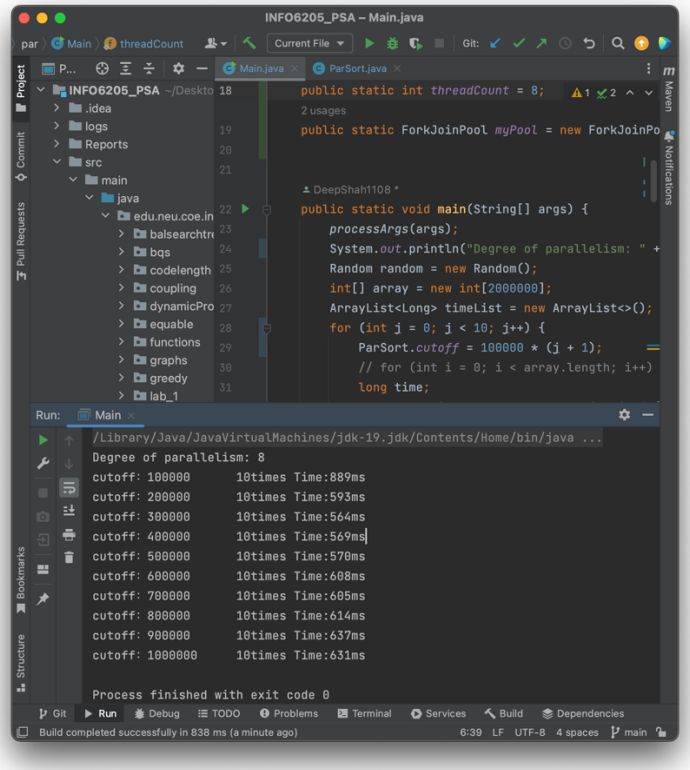
Cutoff Ratio	Thread 2	Thread 4	Thread 8	Thread 16	Thread 32
0.05	1001	926	889	909	1025
0.1	785	632	593	598	569
0.15	796	644	564	581	592
0.2	792	645	569	570	574
0.25	818	647	570	563	560
0.3	870	615	608	604	605
0.35	861	633	605	607	621
0.4	890	615	614	616	616
0.45	864	633	637	627	610
0.5	867	613	631	602	635

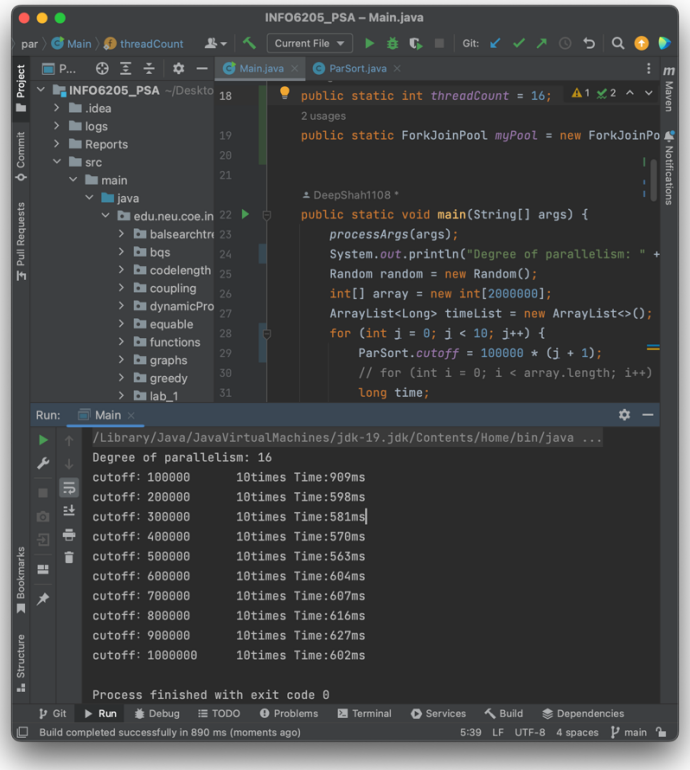

Relationship between cutoff ratio vs different thread values for sorting an array size of 2 million
1200
1000
800
600
400
200
0
0
0.1
0.2
0.3
0.4
0.5
0.6
Thread 2
Thread 4
Thread 8
Thread 16
Thread 32

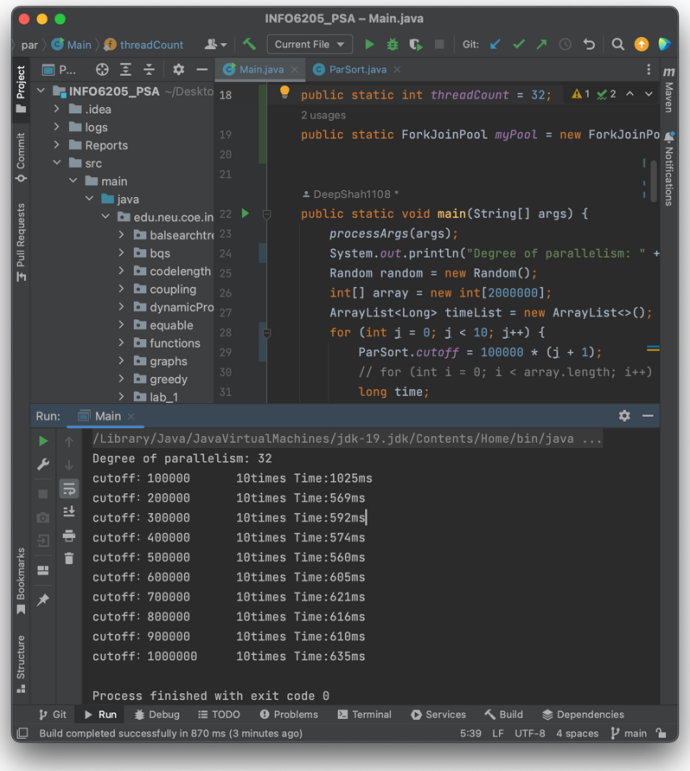
Output:


INFO6205_PSA – Main.java
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 2
cutoff: 100000 10times Time:1001ms
cutoff: 200000 10times Time:785ms
cutoff: 300000 10times Time:796ms
cutoff: 400000 10times Time:792ms
cutoff: 500000 10times Time:818ms
cutoff: 600000 10times Time:870ms
cutoff: 700000 10times Time:861ms
cutoff: 800000 10times Time:890ms
cutoff: 900000 10times Time:864ms
cutoff: 1000000 10times Time:867ms
Process finished with exit code 0


INFO6205_PSA – Main.java
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 4
cutoff: 100000 10times Time:926ms
cutoff: 200000 10times Time:632ms
cutoff: 300000 10times Time:644ms
cutoff: 400000 10times Time:645ms
cutoff: 500000 10times Time:647ms
cutoff: 600000 10times Time:615ms
cutoff: 700000 10times Time:633ms
cutoff: 800000 10times Time:615ms
cutoff: 900000 10times Time:633ms
cutoff: 1000000 10times Time:613ms
Process finished with exit code 0


INFO6205_PSA – Main.java
public static int threadCount = 8;
2 usages
public static ForkJoinPool myPool = new ForkJoinPo
DeepShah1108 *
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[2000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 100000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 8
cutoff: 100000 10times Time:889ms
cutoff: 200000 10times Time:593ms
cutoff: 300000 10times Time:564ms
cutoff: 400000 10times Time:569ms
cutoff: 500000 10times Time:570ms
cutoff: 600000 10times Time:608ms
cutoff: 700000 10times Time:605ms
cutoff: 800000 10times Time:614ms
cutoff: 900000 10times Time:637ms
cutoff: 1000000 10times Time:631ms
Process finished with exit code 0
Build completed successfully in 838 ms (a minute ago)

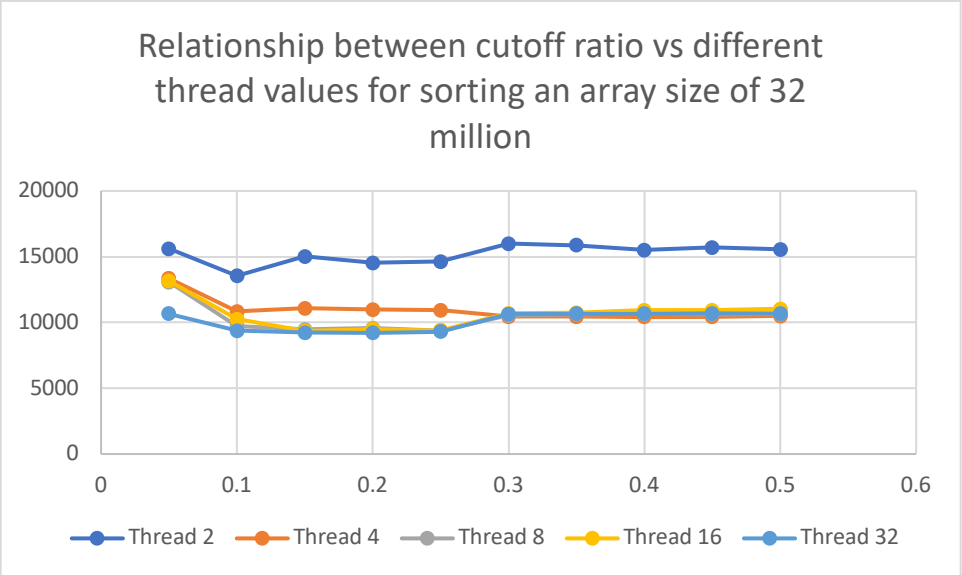

INFO6205_PSA – Main.java
public static int threadCount = 16;
2 usages
public static ForkJoinPool myPool = new ForkJoinPo
DeepShah1108 *
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[2000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 100000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 16
cutoff: 100000 10times Time:909ms
cutoff: 200000 10times Time:598ms
cutoff: 300000 10times Time:581ms
cutoff: 400000 10times Time:570ms
cutoff: 500000 10times Time:563ms
cutoff: 600000 10times Time:604ms
cutoff: 700000 10times Time:607ms
cutoff: 800000 10times Time:616ms
cutoff: 900000 10times Time:627ms
cutoff: 1000000 10times Time:602ms
Process finished with exit code 0
Build completed successfully in 890 ms (moments ago)


INFO6205_PSA – Main.java
public static int threadCount = 32;
2 usages
public static ForkJoinPool myPool = new ForkJoinPo
DeepShah1108 *
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[2000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 100000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 32
cutoff: 100000 10times Time:1025ms
cutoff: 200000 10times Time:569ms
cutoff: 300000 10times Time:592ms
cutoff: 400000 10times Time:574ms
cutoff: 500000 10times Time:560ms
cutoff: 600000 10times Time:605ms
cutoff: 700000 10times Time:621ms
cutoff: 800000 10times Time:616ms
cutoff: 900000 10times Time:610ms
cutoff: 1000000 10times Time:635ms
Process finished with exit code 0
Build completed successfully in 870 ms (3 minutes ago)

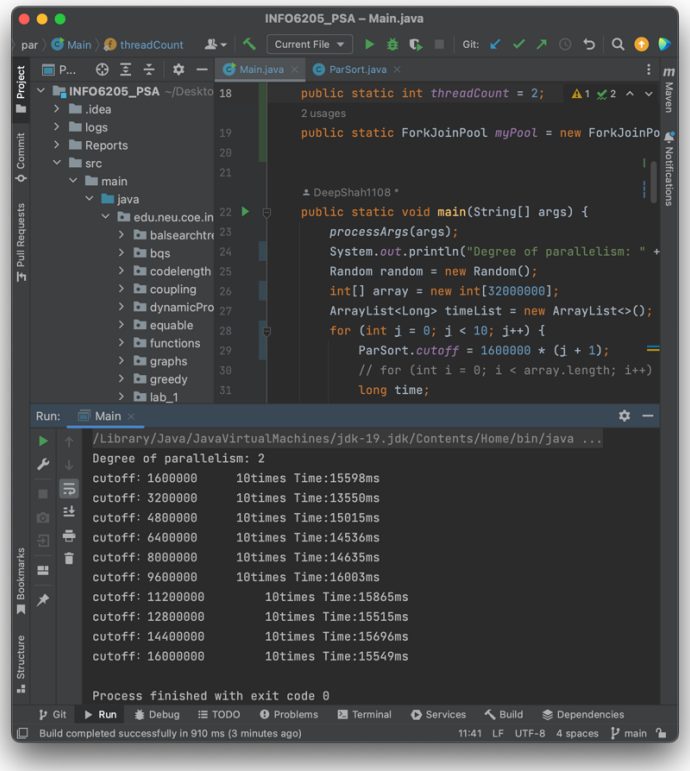
Evidence to Support that conclusion:

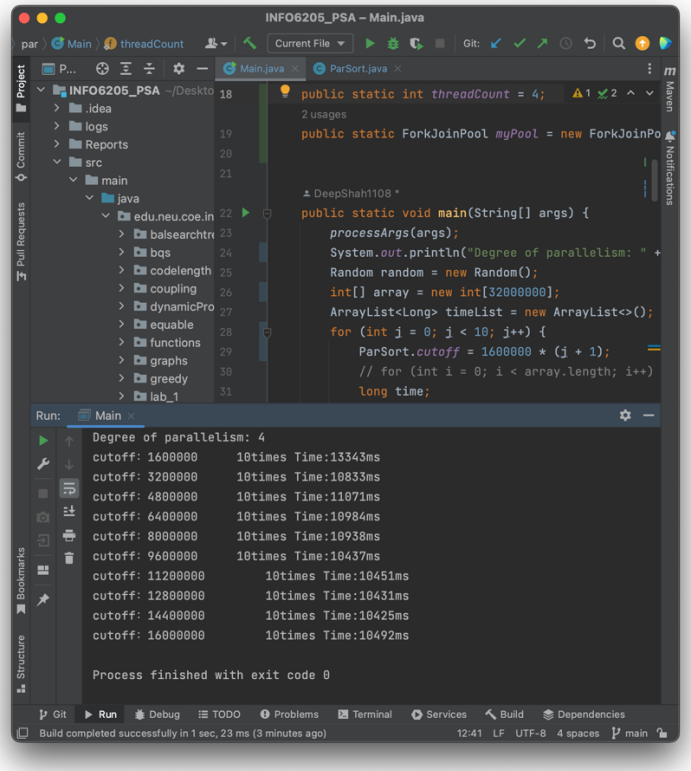
Varying thread from 2 to 32, for array size 32million with cutoff ration from 0.05 to 0.5

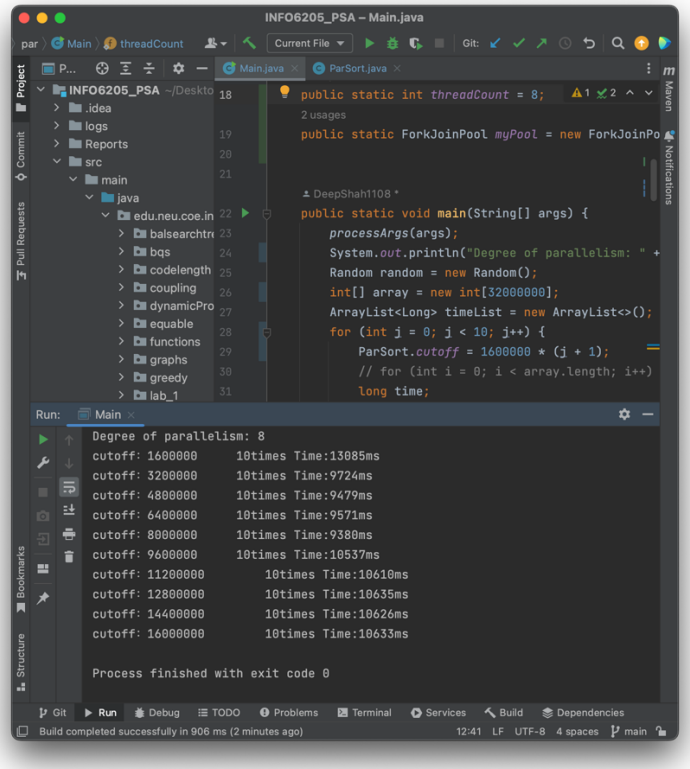
Cutoff Ratio	Thread 2	Thread 4	Thread 8	Thread 16	Thread 32
0.05	15598	13343	13085	13158	10670
0.1	13550	10833	9724	10248	9364
0.15	15015	11071	9479	9362	9222
0.2	14536	10984	9571	9494	9208
0.25	14635	10938	9380	9411	9308
0.3	16003	10437	10537	10691	10648
0.35	15865	10451	10610	10743	10670
0.4	15515	10431	10635	10940	10665
0.45	15696	10425	10626	10928	10698
0.5	15549	10492	10633	11023	10677

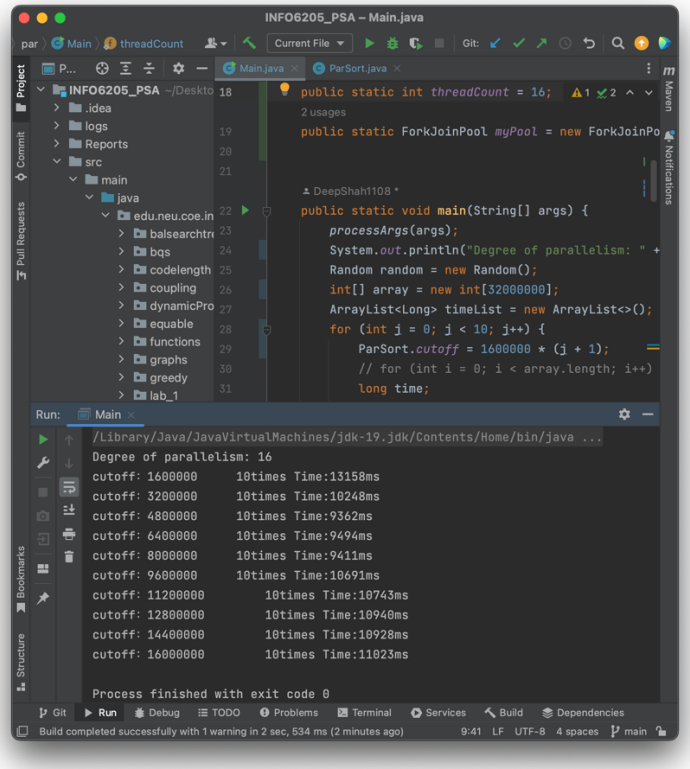


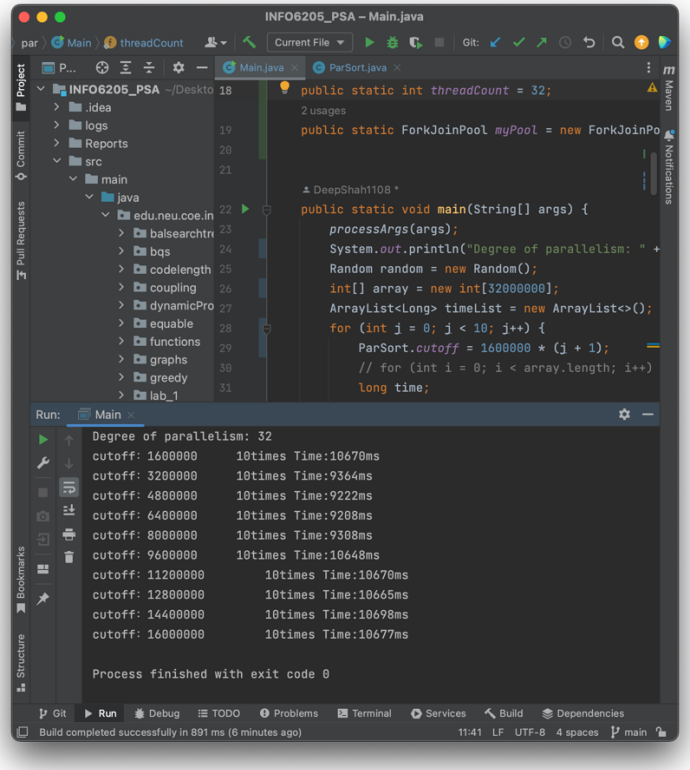
Output:


INFO6205_PSA – Main.java
public static int threadCount = 2;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 2
cutoff: 1600000 10times Time:15598ms
cutoff: 3200000 10times Time:13550ms
cutoff: 4800000 10times Time:15015ms
cutoff: 6400000 10times Time:14536ms
cutoff: 8000000 10times Time:14635ms
cutoff: 9600000 10times Time:16003ms
cutoff: 11200000 10times Time:15865ms
cutoff: 12800000 10times Time:15515ms
cutoff: 14400000 10times Time:15696ms
cutoff: 16000000 10times Time:15549ms
Process finished with exit code 0
Build completed successfully in 910 ms (3 minutes ago)


INFO6205_PSA – Main.java
public static int threadCount = 4;
Run: Main
Degree of parallelism: 4
cutoff: 1600000 10times Time:13343ms
cutoff: 3200000 10times Time:10833ms
cutoff: 4800000 10times Time:11071ms
cutoff: 6400000 10times Time:10984ms
cutoff: 8000000 10times Time:10938ms
cutoff: 9600000 10times Time:10437ms
cutoff: 11200000 10times Time:10451ms
cutoff: 12800000 10times Time:10431ms
cutoff: 14400000 10times Time:10425ms
cutoff: 16000000 10times Time:10492ms
Process finished with exit code 0
Build completed successfully in 1 sec, 23 ms (3 minutes ago)


INFO6205_PSA – Main.java
public static int threadCount = 8;
Run: Main
Degree of parallelism: 8
cutoff: 1600000 10times Time:13085ms
cutoff: 3200000 10times Time:9724ms
cutoff: 4800000 10times Time:9479ms
cutoff: 6400000 10times Time:9571ms
cutoff: 8000000 10times Time:9380ms
cutoff: 9600000 10times Time:10537ms
cutoff: 11200000 10times Time:10610ms
cutoff: 12800000 10times Time:10635ms
cutoff: 14400000 10times Time:10626ms
cutoff: 16000000 10times Time:10633ms
Process finished with exit code 0
Build completed successfully in 906 ms (2 minutes ago)


INFO6205_PSA – Main.java
public static int threadCount = 16;
Run: Main
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
Degree of parallelism: 16
cutoff: 1600000 10times Time:13158ms
cutoff: 3200000 10times Time:10248ms
cutoff: 4800000 10times Time:9362ms
cutoff: 6400000 10times Time:9494ms
cutoff: 8000000 10times Time:9411ms
cutoff: 9600000 10times Time:10691ms
cutoff: 11200000 10times Time:10743ms
cutoff: 12800000 10times Time:10940ms
cutoff: 14400000 10times Time:10928ms
cutoff: 16000000 10times Time:11023ms
Process finished with exit code 0
Build completed successfully with 1 warning in 2 sec, 534 ms (2 minutes ago)


INFO6205_PSA – Main.java
Main.java
ParSort.java
public static int threadCount = 32;
2 usages
public static ForkJoinPool myPool = new ForkJoinPo
DeepShah1108 *
public static void main(String[] args) {
processArgs(args);
System.out.println("Degree of parallelism: " +
Random random = new Random();
int[] array = new int[32000000];
ArrayList<Long> timeList = new ArrayList<>();
for (int j = 0; j < 10; j++) {
ParSort.cutoff = 1600000 * (j + 1);
// for (int i = 0; i < array.length; i++)
long time;
Run: Main
Degree of parallelism: 32
cutoff: 1600000 10times Time:10670ms
cutoff: 3200000 10times Time:9364ms
cutoff: 4800000 10times Time:9222ms
cutoff: 6400000 10times Time:9208ms
cutoff: 8000000 10times Time:9308ms
cutoff: 9600000 10times Time:10648ms
cutoff: 11200000 10times Time:10670ms
cutoff: 12800000 10times Time:10665ms
cutoff: 14400000 10times Time:10698ms
cutoff: 16000000 10times Time:10677ms
Process finished with exit code 0
Build completed successfully in 891 ms (6 minutes ago)