Venkata Raghu Teja Kumar Gollapudi (001529656)

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.

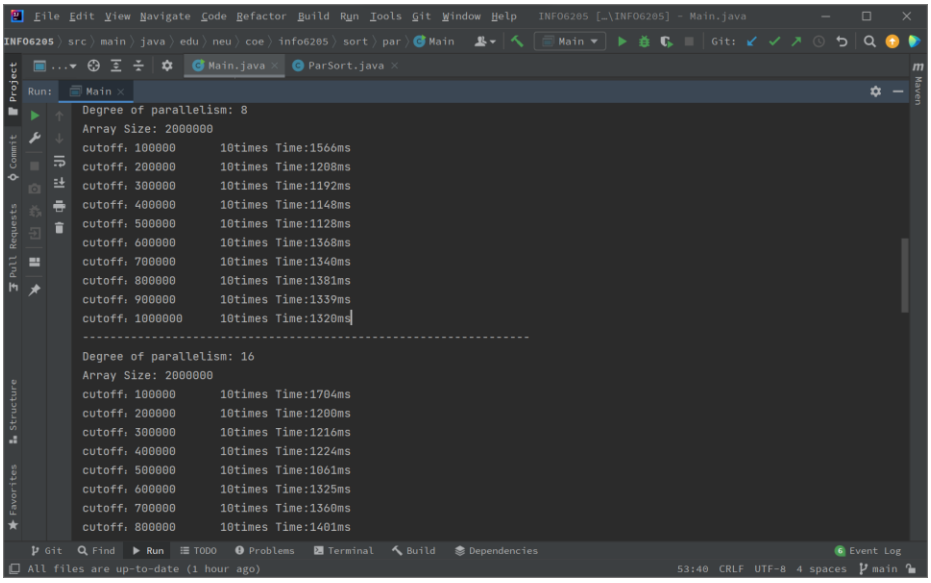
Output

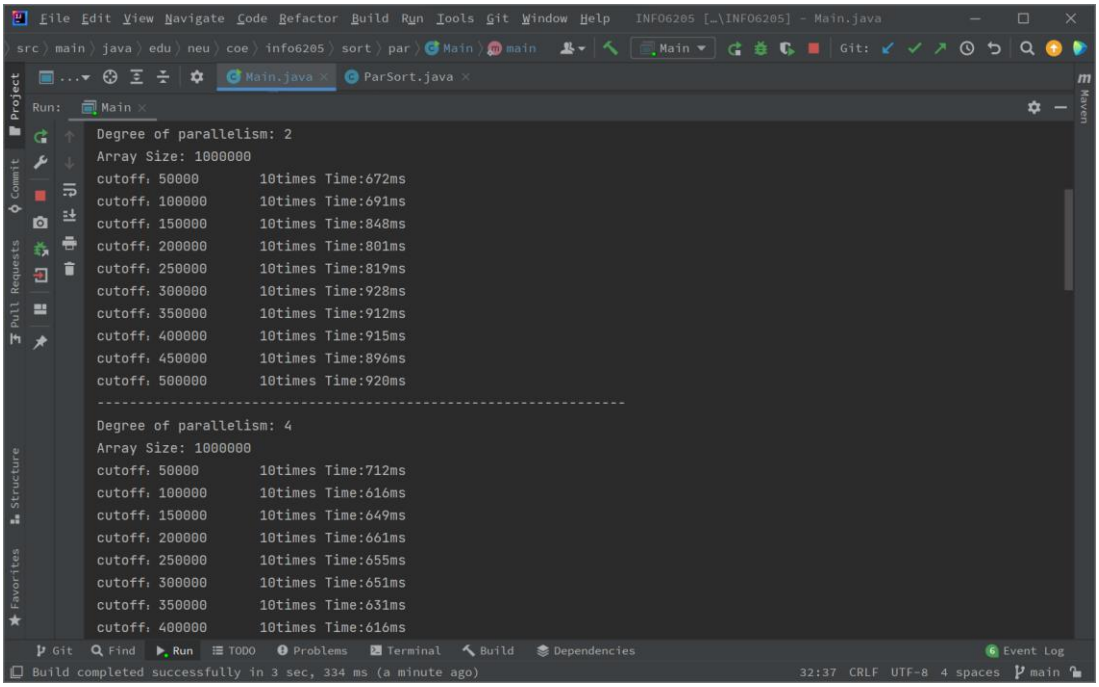
Here, I set the degree of parallelism in the multiples of 2 while keeping the base as 1. For the experiments, I kept changing the cutoff values and the array sizes to test the sorting to its optimal level.

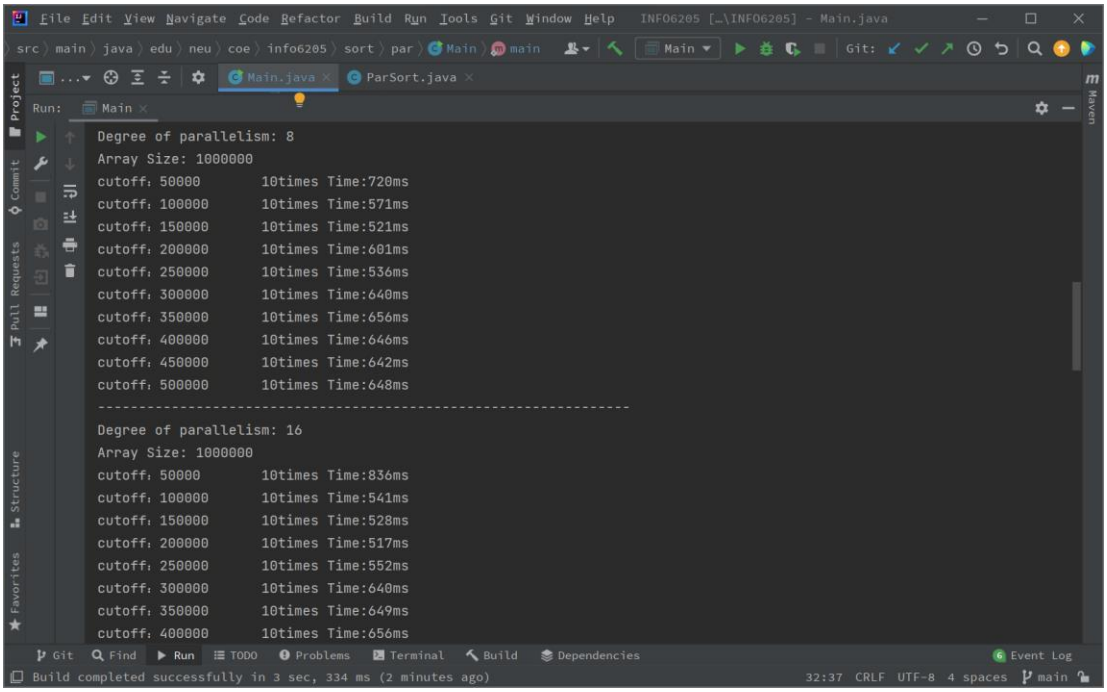
I have used 3 different array sizes of length 500000, 1 million and 2 million.

```
Degree of parallelism: 1
Array Size: 2000000
cutoff: 100000      10times Time:2482ms
cutoff: 200000      10times Time:1804ms
cutoff: 300000      10times Time:2226ms
cutoff: 400000      10times Time:2199ms
cutoff: 500000      10times Time:2305ms
cutoff: 600000      10times Time:3050ms
cutoff: 700000      10times Time:2883ms
cutoff: 800000      10times Time:2892ms
cutoff: 900000      10times Time:2821ms
cutoff: 1000000     10times Time:2808ms
```

```
Degree of parallelism: 2
Array Size: 2000000
cutoff: 100000      10times Time:1513ms
cutoff: 200000      10times Time:1511ms
cutoff: 300000      10times Time:1705ms
cutoff: 400000      10times Time:1653ms
cutoff: 500000      10times Time:1746ms
cutoff: 600000      10times Time:1994ms
cutoff: 700000      10times Time:1868ms
cutoff: 800000      10times Time:2114ms
cutoff: 900000      10times Time:1760ms
cutoff: 1000000     10times Time:1926ms
---------------------------------------------------------------------
Degree of parallelism: 4
Array Size: 2000000
cutoff: 100000      10times Time:1712ms
cutoff: 200000      10times Time:1369ms
cutoff: 300000      10times Time:1458ms
cutoff: 400000      10times Time:1436ms
cutoff: 500000      10times Time:1457ms
cutoff: 600000      10times Time:1368ms
cutoff: 700000      10times Time:1401ms
cutoff: 800000      10times Time:1360ms
```




File Edit View Navigate Code Refactor Build Run Tools Git Window Help
INFO6205 [...\INFO6205] - Main.java
Main.java
ParSort.java
Run: Main
Degree of parallelism: 2
Array Size: 1000000
cutoff: 50000 10times Time:672ms
cutoff: 100000 10times Time:691ms
cutoff: 150000 10times Time:848ms
cutoff: 200000 10times Time:801ms
cutoff: 250000 10times Time:819ms
cutoff: 300000 10times Time:928ms
cutoff: 350000 10times Time:912ms
cutoff: 400000 10times Time:915ms
cutoff: 450000 10times Time:896ms
cutoff: 500000 10times Time:920ms
Degree of parallelism: 4
Array Size: 1000000
cutoff: 50000 10times Time:712ms
cutoff: 100000 10times Time:616ms
cutoff: 150000 10times Time:649ms
cutoff: 200000 10times Time:661ms
cutoff: 250000 10times Time:655ms
cutoff: 300000 10times Time:651ms
cutoff: 350000 10times Time:631ms
cutoff: 400000 10times Time:616ms
Git Find Run TODO Problems Terminal Build Dependencies
Event Log
Build completed successfully in 3 sec, 334 ms (a minute ago)
32:37 CRLF UTF-8 4 spaces main


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
INFO6205 [...\INFO6205] - Main.java
Main.java
ParSort.java
Run: Main
Degree of parallelism: 8
Array Size: 1000000
cutoff: 50000 10times Time:720ms
cutoff: 100000 10times Time:571ms
cutoff: 150000 10times Time:521ms
cutoff: 200000 10times Time:601ms
cutoff: 250000 10times Time:536ms
cutoff: 300000 10times Time:640ms
cutoff: 350000 10times Time:656ms
cutoff: 400000 10times Time:646ms
cutoff: 450000 10times Time:642ms
cutoff: 500000 10times Time:648ms
Degree of parallelism: 16
Array Size: 1000000
cutoff: 50000 10times Time:836ms
cutoff: 100000 10times Time:541ms
cutoff: 150000 10times Time:528ms
cutoff: 200000 10times Time:517ms
cutoff: 250000 10times Time:552ms
cutoff: 300000 10times Time:640ms
cutoff: 350000 10times Time:649ms
cutoff: 400000 10times Time:656ms
Git Find Run TODO Problems Terminal Build Dependencies
Event Log
Build completed successfully in 3 sec, 334 ms (2 minutes ago)
32:37 CRLF UTF-8 4 spaces main

Arraysize = 500000

Thread 1		Thread 2		Thread 4		Thread 8		Thread 16		Thread 32		Thread 64		Thread 128		Thread 256	
Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time
20000	1033	20000	400	20000	432	20000	352	20000	400	20000	300	20000	280	20000	288	20000	320
40000	444	40000	368	40000	336	40000	280	40000	264	40000	268	40000	272	40000	272	40000	272
60000	435	60000	376	60000	328	60000	288	60000	273	60000	267	60000	272	60000	265	60000	264
80000	547	80000	449	80000	351	80000	272	80000	264	80000	268	80000	272	80000	264	80000	280
100000	548	100000	432	100000	343	100000	264	100000	272	100000	272	100000	272	100000	264	100000	272
120000	535	120000	419	120000	352	120000	272	120000	272	120000	263	120000	280	120000	256	120000	274
140000	650	140000	490	140000	328	140000	318	140000	312	140000	319	140000	308	140000	312	140000	314
160000	665	160000	456	160000	329	160000	333	160000	328	160000	328	160000	320	160000	320	160000	329
180000	640	180000	472	180000	320	180000	307	180000	325	180000	320	180000	329	180000	312	180000	321
200000	672	200000	464	200000	320	200000	318	200000	318	200000	304	200000	336	200000	328	200000	337

Arraysize = 1000000

Thread 1		Thread 2		Thread 4		Thread 8		Thread 16		Thread 32		Thread 64		Thread 128		Thread 256	
Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time
50000	1442	50000	744	50000	779	50000	839	50000	768	50000	589	50000	576	50000	536	50000	568
100000	892	100000	768	100000	654	100000	565	100000	528	100000	556	100000	560	100000	536	100000	534
150000	1080	150000	856	150000	696	150000	560	150000	520	150000	562	150000	529	150000	576	150000	544
200000	1105	200000	880	200000	712	200000	618	200000	560	200000	520	200000	526	200000	568	200000	528
250000	1104	250000	864	250000	704	250000	578	250000	573	250000	536	250000	520	250000	521	250000	544
300000	1296	300000	944	300000	664	300000	623	300000	641	300000	632	300000	640	300000	647	300000	655
350000	1384	350000	953	350000	672	350000	624	350000	653	350000	650	350000	648	350000	632	350000	649
400000	1296	400000	964	400000	689	400000	627	400000	640	400000	641	400000	656	400000	648	400000	631
450000	1279	450000	944	450000	672	450000	669	450000	652	450000	648	450000	657	450000	632	450000	637
500000	1335	500000	936	500000	664	500000	632	500000	648	500000	640	500000	643	500000	648	500000	656

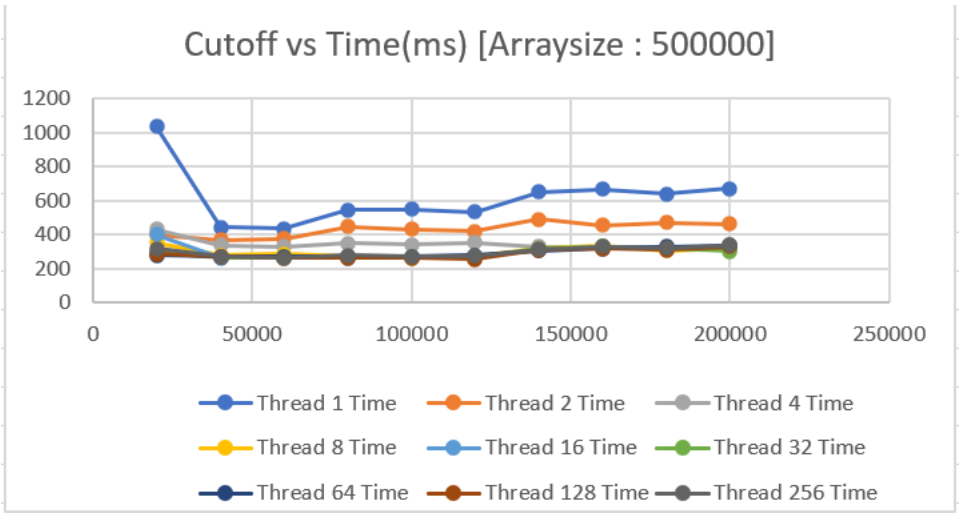
Arraysize = 2000000

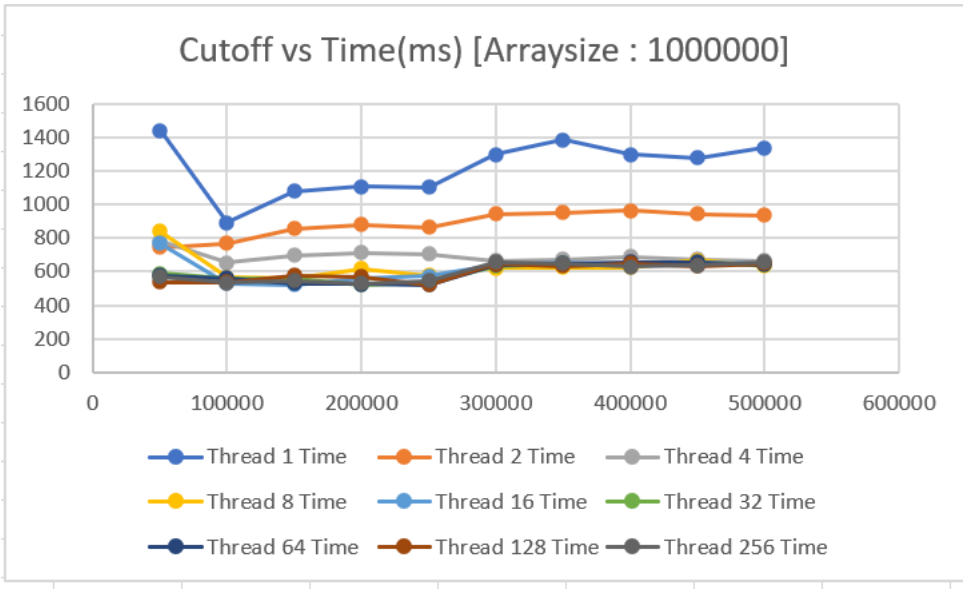
Thread 1		Thread 2		Thread 4		Thread 8		Thread 16		Thread 32		Thread 64		Thread 128		Thread 256	
Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time	Cutoff	Time
100000	2412	100000	1897	100000	1512	100000	1640	100000	1728	100000	1205	100000	1181	100000	1203	100000	1193
200000	1860	200000	1549	200000	1364	200000	1289	200000	1264	200000	1272	200000	1158	200000	1255	200000	1200
300000	2313	300000	1760	300000	1504	300000	1200	300000	1429	300000	1176	300000	1142	300000	1081	300000	1194
400000	2360	400000	1840	400000	1489	400000	1088	400000	1223	400000	1072	400000	1140	400000	1208	400000	1180
500000	2297	500000	1809	500000	1481	500000	1160	500000	1144	500000	1191	500000	1189	500000	1151	500000	1233
600000	3003	600000	2021	600000	1393	600000	1346	600000	1385	600000	1335	600000	1351	600000	1384	600000	1369
700000	3066	700000	2027	700000	1440	700000	1363	700000	1361	700000	1360	700000	1375	700000	1384	700000	1388
800000	2937	800000	2021	800000	1472	800000	1392	800000	1347	800000	1336	800000	1344	800000	1363	800000	1369
900000	2860	900000	2052	900000	1365	900000	1360	900000	1384	900000	1366	900000	1356	900000	1380	900000	1376
1000000	3057	1000000	2000	1000000	1376	1000000	1355	1000000	1371	1000000	1397	1000000	1361	1000000	1392	1000000	1393

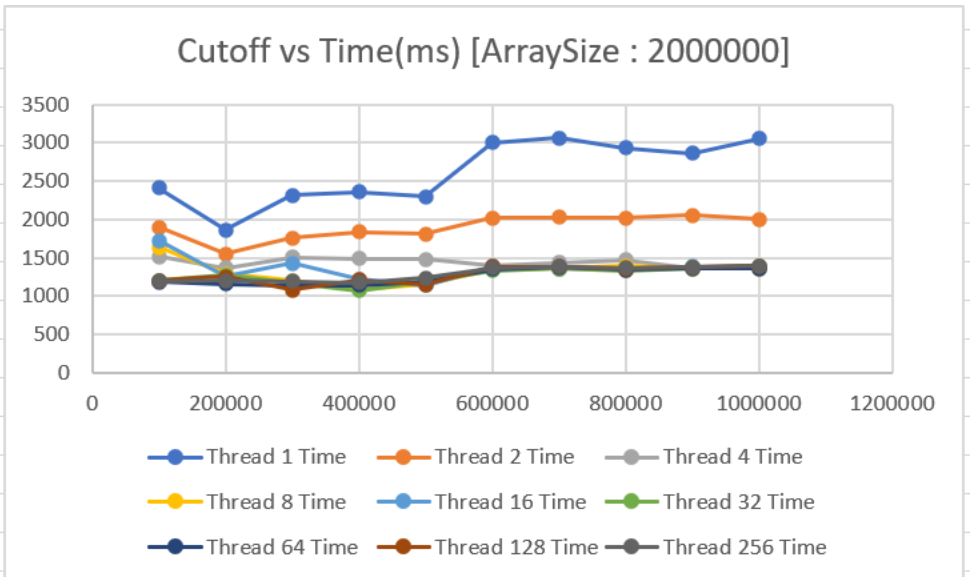
The number of threads are taken as the powers of 2 since it is decided by recursion depth.

Relationship Conclusion and Evidence

The cutoff-time chart are as follows:


Cutoff vs Time(ms) [Arraysize : 500000]
1200
1000
800
600
400
200
0
0
50000
100000
150000
200000
250000
Thread 1 Time
Thread 2 Time
Thread 4 Time
Thread 8 Time
Thread 16 Time
Thread 32 Time
Thread 64 Time
Thread 128 Time
Thread 256 Time


Cutoff vs Time(ms) [Arraysize : 1000000]
1600
1400
1200
1000
800
600
400
200
0
0
100000
200000
300000
400000
500000
600000
Thread 1 Time
Thread 2 Time
Thread 4 Time
Thread 8 Time
Thread 16 Time
Thread 32 Time
Thread 64 Time
Thread 128 Time
Thread 256 Time



From the above charts, I have observed that as the cutoff increases, the time drastically decreases at the beginning and then increases. Therefore, it can be concluded that the lowest time is about where cutoff is around **30% of the array size**.

The **optimum number of threads is 8** after which the increase in threads doesn't provide better output.