Ankita Senapati (001003695)

Program Structures & Algorithms

Fall 2021

Assignment No. 5

Task

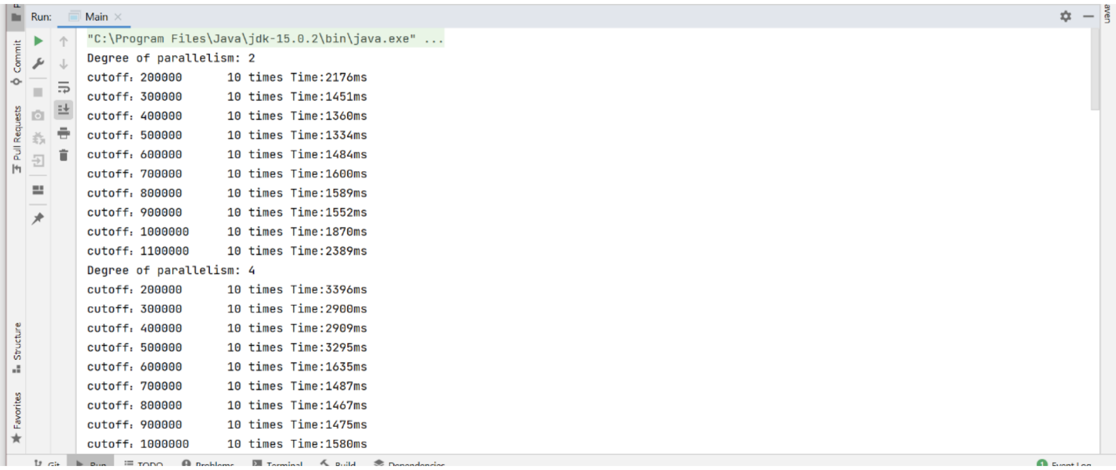
Your task is to implement a parallel sorting algorithm such that each partition of the array is sorted in parallel. You will consider two different schemes for deciding whether to sort in parallel.

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.

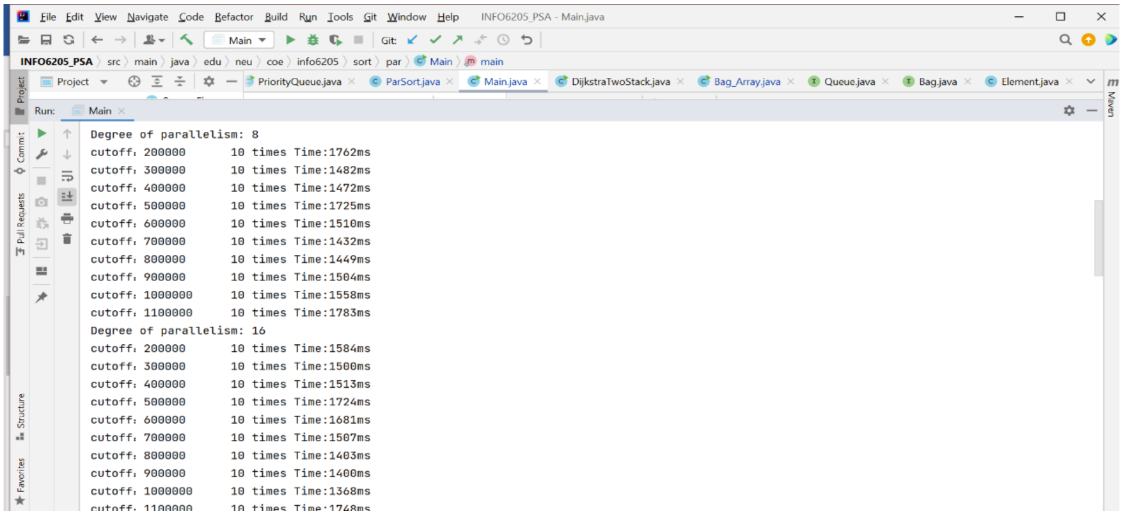
Output

Adhering to the demands of the task, I have stuck to the powers of 2 for the size of thread. With the peak thread count of 1024, I have performed the experiments extensively. I have also used 3 different array sizes with varying cutoffs.

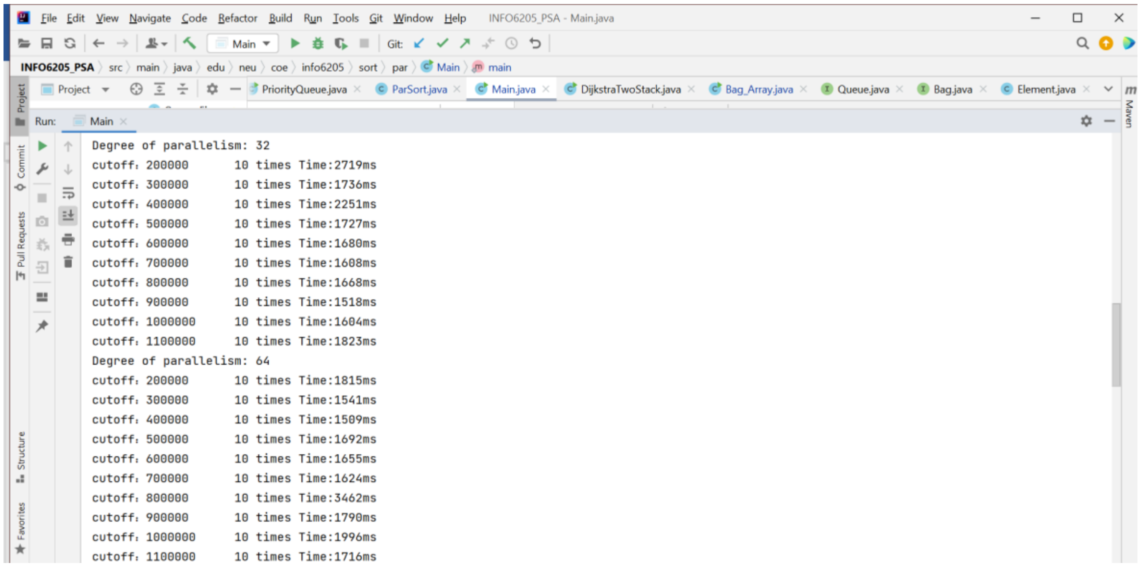
Output for Arraysize : 200000.



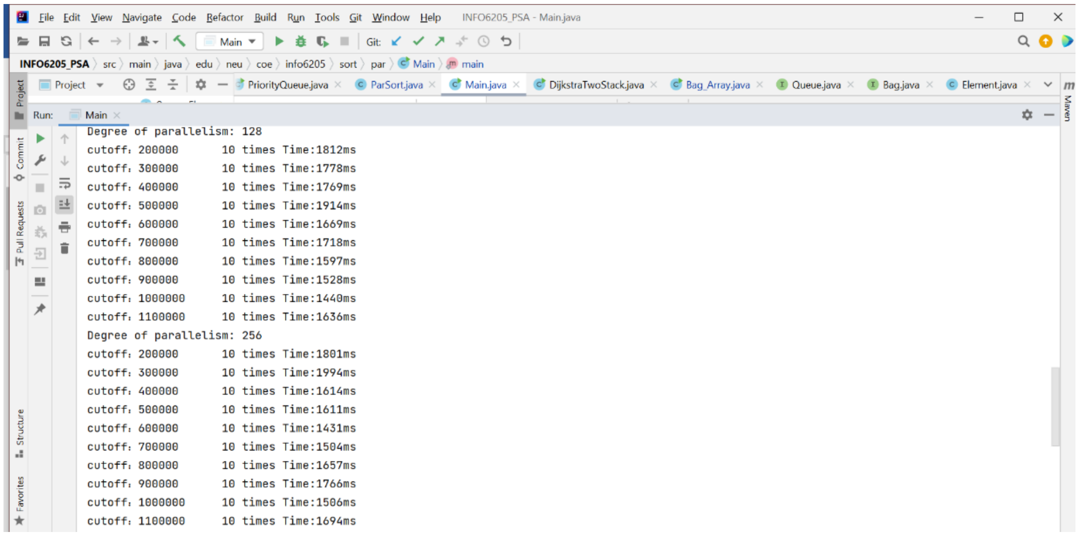
```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" ...
Degree of parallelism: 2
cutoff：200000      10 times Time:2176ms
cutoff：300000      10 times Time:1451ms
cutoff：400000      10 times Time:1360ms
cutoff：500000      10 times Time:1334ms
cutoff：600000      10 times Time:1484ms
cutoff：700000      10 times Time:1600ms
cutoff：800000      10 times Time:1589ms
cutoff：900000      10 times Time:1552ms
cutoff：1000000     10 times Time:1870ms
cutoff：1100000     10 times Time:2389ms
Degree of parallelism: 4
cutoff：200000      10 times Time:3396ms
cutoff：300000      10 times Time:2900ms
cutoff：400000      10 times Time:2909ms
cutoff：500000      10 times Time:3295ms
cutoff：600000      10 times Time:1635ms
cutoff：700000      10 times Time:1487ms
cutoff：800000      10 times Time:1467ms
cutoff：900000      10 times Time:1475ms
cutoff：1000000     10 times Time:1580ms
```


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
INFO6205_PSA - Main.java
Main
Git:
INFO6205_PSA › src › main › java › edu › neu › coe › info6205 › sort › par › Main › main
Project
PriorityQueue.java
ParSort.java
Main.java
DijkstraTwoStack.java
Bag_Array.java
Queue.java
Bag.java
Element.java
Run:
Main
Maven

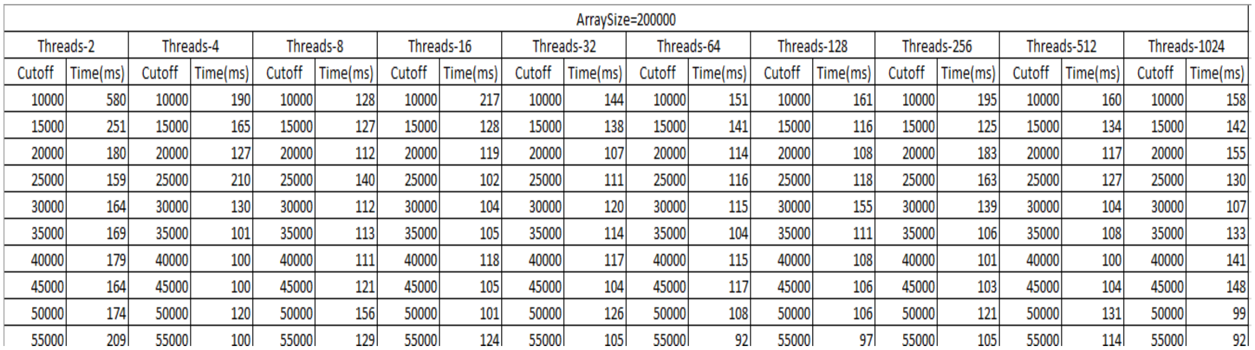
```
Degree of parallelism: 8
cutoff：200000      10 times Time:1762ms
cutoff：300000      10 times Time:1482ms
cutoff：400000      10 times Time:1472ms
cutoff：500000      10 times Time:1725ms
cutoff：600000      10 times Time:1510ms
cutoff：700000      10 times Time:1432ms
cutoff：800000      10 times Time:1449ms
cutoff：900000      10 times Time:1504ms
cutoff：1000000     10 times Time:1558ms
cutoff：1100000     10 times Time:1783ms
Degree of parallelism: 16
cutoff：200000      10 times Time:1584ms
cutoff：300000      10 times Time:1500ms
cutoff：400000      10 times Time:1513ms
cutoff：500000      10 times Time:1724ms
cutoff：600000      10 times Time:1681ms
cutoff：700000      10 times Time:1507ms
cutoff：800000      10 times Time:1403ms
cutoff：900000      10 times Time:1400ms
cutoff：1000000     10 times Time:1368ms
cutoff：1100000     10 times Time:1748ms
```


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
INFO6205_PSA - Main.java
Main
Git:
INFO6205_PSA › src › main › java › edu › neu › coe › info6205 › sort › par › Main › main
Project
PriorityQueue.java
ParSort.java
Main.java
DijkstraTwoStack.java
Bag_Array.java
Queue.java
Bag.java
Element.java
Run:
Main
Maven

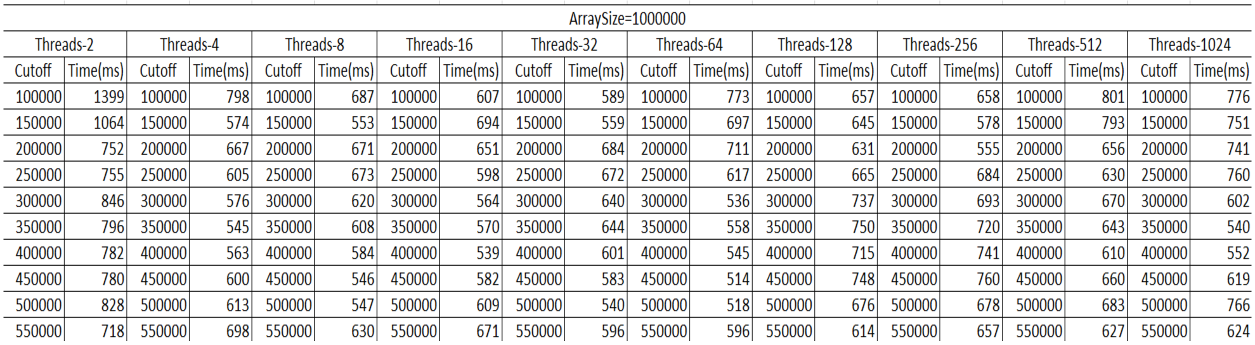
```
Degree of parallelism: 32
cutoff：200000      10 times Time:2719ms
cutoff：300000      10 times Time:1736ms
cutoff：400000      10 times Time:2251ms
cutoff：500000      10 times Time:1727ms
cutoff：600000      10 times Time:1680ms
cutoff：700000      10 times Time:1608ms
cutoff：800000      10 times Time:1668ms
cutoff：900000      10 times Time:1518ms
cutoff：1000000     10 times Time:1604ms
cutoff：1100000     10 times Time:1823ms
Degree of parallelism: 64
cutoff：200000      10 times Time:1815ms
cutoff：300000      10 times Time:1541ms
cutoff：400000      10 times Time:1509ms
cutoff：500000      10 times Time:1692ms
cutoff：600000      10 times Time:1655ms
cutoff：700000      10 times Time:1624ms
cutoff：800000      10 times Time:3462ms
cutoff：900000      10 times Time:1790ms
cutoff：1000000     10 times Time:1996ms
cutoff：1100000     10 times Time:1716ms
```


File Edit View Navigate Code Refactor Build Run Tools Git Window Help INFO6205_PSA - Main.java
Main
Git:
INFO6205_PSA 〉 src 〉 main 〉 java 〉 edu 〉 neu 〉 coe 〉 info6205 〉 sort 〉 par 〉 Main 〉 main
Project
PriorityQueue.java
ParSort.java
Main.java
DijkstraTwoStack.java
Bag_Array.java
Queue.java
Bag.java
Element.java
Run: Main
Project
Commit
Pull Requests
Structure
Favorites
Maven

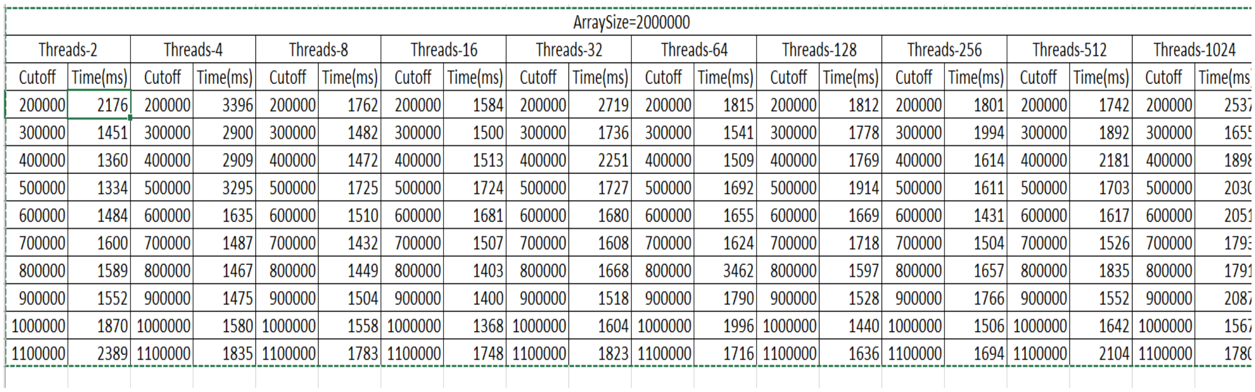
```
Degree of parallelism: 128
cutoff: 200000      10 times Time:1812ms
cutoff: 300000      10 times Time:1778ms
cutoff: 400000      10 times Time:1769ms
cutoff: 500000      10 times Time:1914ms
cutoff: 600000      10 times Time:1669ms
cutoff: 700000      10 times Time:1718ms
cutoff: 800000      10 times Time:1597ms
cutoff: 900000      10 times Time:1528ms
cutoff: 1000000     10 times Time:1440ms
cutoff: 1100000     10 times Time:1636ms
Degree of parallelism: 256
cutoff: 200000      10 times Time:1801ms
cutoff: 300000      10 times Time:1994ms
cutoff: 400000      10 times Time:1614ms
cutoff: 500000      10 times Time:1611ms
cutoff: 600000      10 times Time:1431ms
cutoff: 700000      10 times Time:1504ms
cutoff: 800000      10 times Time:1657ms
cutoff: 900000      10 times Time:1766ms
cutoff: 1000000     10 times Time:1506ms
cutoff: 1100000     10 times Time:1694ms
```



ArraySize=200000																			
Threads-2		Threads-4		Threads-8		Threads-16		Threads-32		Threads-64		Threads-128		Threads-256		Threads-512		Threads-1024	
Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)
10000	580	10000	190	10000	128	10000	217	10000	144	10000	151	10000	161	10000	195	10000	160	10000	158
15000	251	15000	165	15000	127	15000	128	15000	138	15000	141	15000	116	15000	125	15000	134	15000	142
20000	180	20000	127	20000	112	20000	119	20000	107	20000	114	20000	108	20000	183	20000	117	20000	155
25000	159	25000	210	25000	140	25000	102	25000	111	25000	116	25000	118	25000	163	25000	127	25000	130
30000	164	30000	130	30000	112	30000	104	30000	120	30000	115	30000	155	30000	139	30000	104	30000	107
35000	169	35000	101	35000	113	35000	105	35000	114	35000	104	35000	111	35000	106	35000	108	35000	133
40000	179	40000	100	40000	111	40000	118	40000	117	40000	115	40000	108	40000	101	40000	100	40000	141
45000	164	45000	100	45000	121	45000	105	45000	104	45000	117	45000	106	45000	103	45000	104	45000	148
50000	174	50000	120	50000	156	50000	101	50000	126	50000	108	50000	106	50000	121	50000	131	50000	99
55000	209	55000	100	55000	129	55000	124	55000	105	55000	92	55000	97	55000	105	55000	114	55000	92



ArraySize=1000000																			
Threads-2		Threads-4		Threads-8		Threads-16		Threads-32		Threads-64		Threads-128		Threads-256		Threads-512		Threads-1024	
Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)
100000	1399	100000	798	100000	687	100000	607	100000	589	100000	773	100000	657	100000	658	100000	801	100000	776
150000	1064	150000	574	150000	553	150000	694	150000	559	150000	697	150000	645	150000	578	150000	793	150000	751
200000	752	200000	667	200000	671	200000	651	200000	684	200000	711	200000	631	200000	555	200000	656	200000	741
250000	755	250000	605	250000	673	250000	598	250000	672	250000	617	250000	665	250000	684	250000	630	250000	760
300000	846	300000	576	300000	620	300000	564	300000	640	300000	536	300000	737	300000	693	300000	670	300000	602
350000	796	350000	545	350000	608	350000	570	350000	644	350000	558	350000	750	350000	720	350000	643	350000	540
400000	782	400000	563	400000	584	400000	539	400000	601	400000	545	400000	715	400000	741	400000	610	400000	552
450000	780	450000	600	450000	546	450000	582	450000	583	450000	514	450000	748	450000	760	450000	660	450000	619
500000	828	500000	613	500000	547	500000	609	500000	540	500000	518	500000	676	500000	678	500000	683	500000	766
550000	718	550000	698	550000	630	550000	671	550000	596	550000	596	550000	614	550000	657	550000	627	550000	624



ArraySize=2000000																			
Threads-2		Threads-4		Threads-8		Threads-16		Threads-32		Threads-64		Threads-128		Threads-256		Threads-512		Threads-1024	
Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms)	Cutoff	Time(ms
200000	2176	200000	3396	200000	1762	200000	1584	200000	2719	200000	1815	200000	1812	200000	1801	200000	1742	200000	2537
300000	1451	300000	2900	300000	1482	300000	1500	300000	1736	300000	1541	300000	1778	300000	1994	300000	1892	300000	1655
400000	1360	400000	2909	400000	1472	400000	1513	400000	2251	400000	1509	400000	1769	400000	1614	400000	2181	400000	1898
500000	1334	500000	3295	500000	1725	500000	1724	500000	1727	500000	1692	500000	1914	500000	1611	500000	1703	500000	2030
600000	1484	600000	1635	600000	1510	600000	1681	600000	1680	600000	1655	600000	1669	600000	1431	600000	1617	600000	2051
700000	1600	700000	1487	700000	1432	700000	1507	700000	1608	700000	1624	700000	1718	700000	1504	700000	1526	700000	1793
800000	1589	800000	1467	800000	1449	800000	1403	800000	1668	800000	3462	800000	1597	800000	1657	800000	1835	800000	1791
900000	1552	900000	1475	900000	1504	900000	1400	900000	1518	900000	1790	900000	1528	900000	1766	900000	1552	900000	2087
1000000	1870	1000000	1580	1000000	1558	1000000	1368	1000000	1604	1000000	1996	1000000	1440	1000000	1506	1000000	1642	1000000	1567
1100000	2389	1100000	1835	1100000	1783	1100000	1748	1100000	1823	1100000	1716	1100000	1636	1100000	1694	1100000	2104	1100000	1780

The data collected through experiments performed on varying cutoff and array sizes.

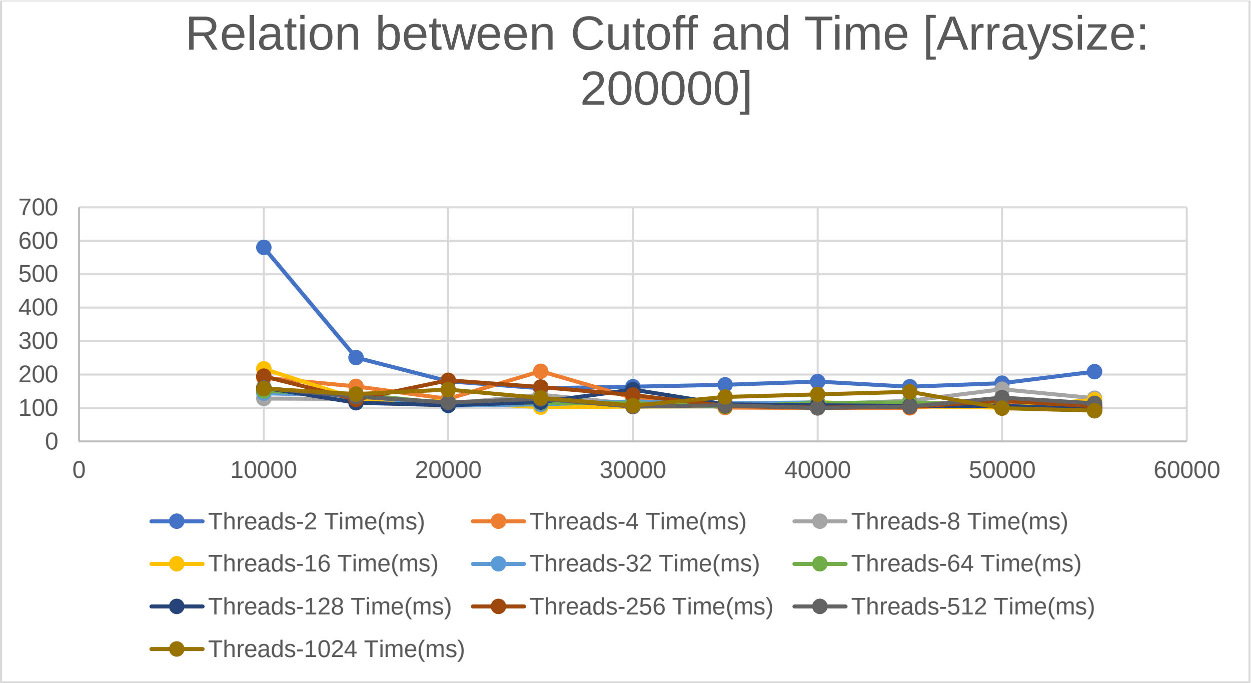
Relationship Conclusion and Evidence

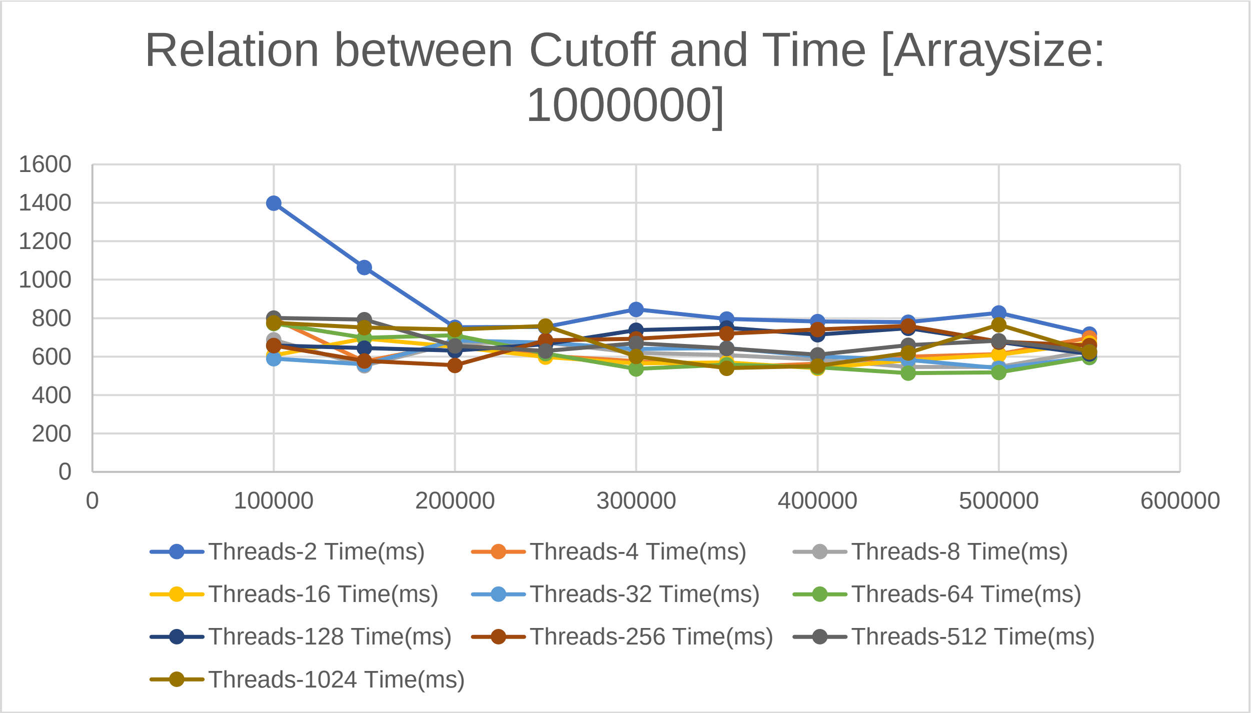
1. Based on the experiments, I have concluded that the optimal cutoff for the best performance of the sort is around 40 % of the array size.

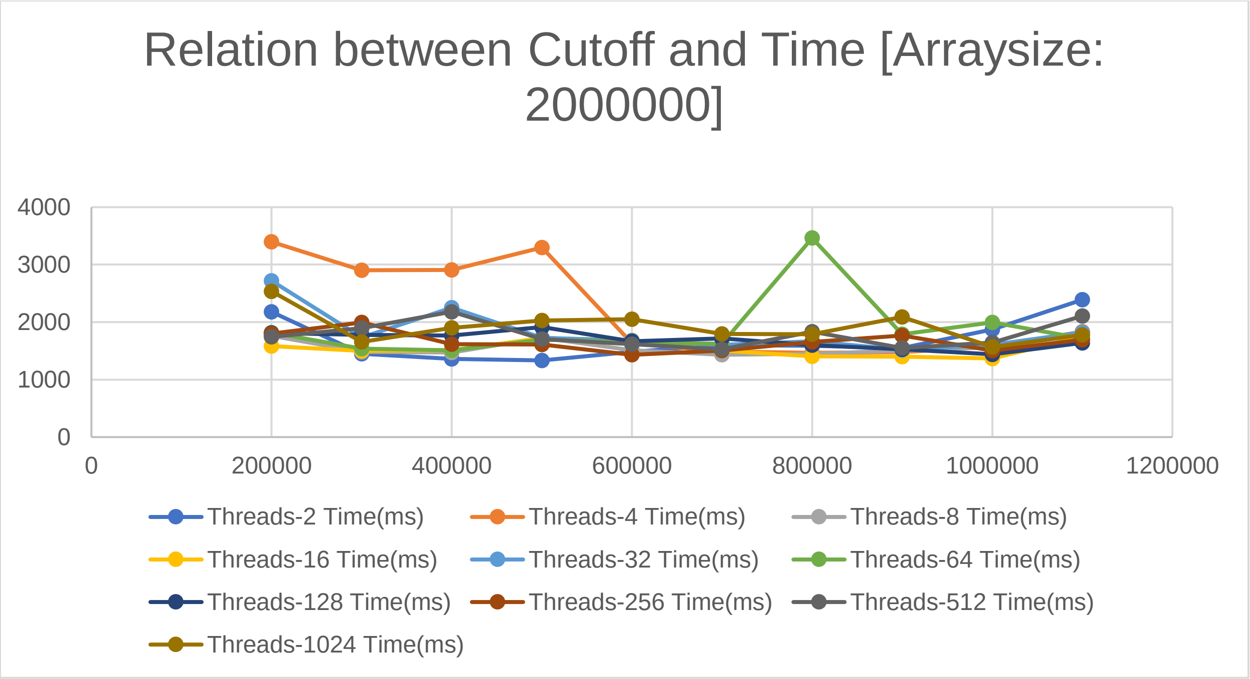
2.The number of threads required for the optimum result is 16. The performance of the sort reduces as the number of threads increase.

One peculiar aspect is that at the start of the sorting, the time reduces drastically and slowly rises.

The cutoff-time chart are as follows:


Relation between Cutoff and Time [Arraysize: 200000]
700
600
500
400
300
200
100
0
0
10000
20000
30000
40000
50000
60000
Threads-2 Time(ms)
Threads-4 Time(ms)
Threads-8 Time(ms)
Threads-16 Time(ms)
Threads-32 Time(ms)
Threads-64 Time(ms)
Threads-128 Time(ms)
Threads-256 Time(ms)
Threads-512 Time(ms)
Threads-1024 Time(ms)


Relation between Cutoff and Time [Arraysize: 1000000]
1600
1400
1200
1000
800
600
400
200
0
0
100000
200000
300000
400000
500000
600000
Threads-2 Time(ms)
Threads-4 Time(ms)
Threads-8 Time(ms)
Threads-16 Time(ms)
Threads-32 Time(ms)
Threads-64 Time(ms)
Threads-128 Time(ms)
Threads-256 Time(ms)
Threads-512 Time(ms)
Threads-1024 Time(ms)


Relation between Cutoff and Time [Arraysize: 2000000]
4000
3000
2000
1000
0
0
200000
400000
600000
800000
1000000
1200000
Threads-2 Time(ms)
Threads-4 Time(ms)
Threads-8 Time(ms)
Threads-16 Time(ms)
Threads-32 Time(ms)
Threads-64 Time(ms)
Threads-128 Time(ms)
Threads-256 Time(ms)
Threads-512 Time(ms)
Threads-1024 Time(ms)