Program Structures and Algorithms

Spring 2023(SEC – 1)

NAME: Aishwarya Venkatesan

NUID: 001569213

**Task:**

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.

* 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.
* 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).
* An appropriate combination of these.

**Relationship Conclusion:**

For running the parallel sort for different cutoff values from 5000 to 50000 and varying the number of threads in powers of 2 from 2 to 64 for different array sizes, we can conclude that the best solution is achieved for thread size 4. Performance is achieved when cut off value is 25% of array size

**Evidence to support that conclusion:**

**For array size of 1000000, cutoff values are considered from 5000 to 50000**

**Degree of parallelism :2**

| cut off value | time of execution |
| --- | --- |
| 5000 | 603ms |
| 10000 | 589ms |
| 15000 | 459ms |
| 20000 | 415ms |
| 25000 | 407ms |
| 30000 | 403ms |
| 35000 | 390ms |
| 40000 | 397ms |
| 45000 | 393ms |
| 50000 | 406ms |

**Degree of parallelism : 4**

| cut off value | time of execution |
| --- | --- |
| 5000 | 556ms |
| 10000 | 366ms |
| 15000 | 300ms |
| 20000 | 294ms |
| 25000 | 292ms |
| 30000 | 295ms |
| 35000 | 283ms |
| 40000 | 289ms |
| 45000 | 287ms |
| 50000 | 286ms |

**Degree of parallelism : 8**

| cut off value | time of execution |
| --- | --- |
| 5000 | 534ms |
| 10000 | 427ms |
| 15000 | 252ms |
| 20000 | 250ms |
| 25000 | 250ms |
| 30000 | 246ms |
| 35000 | 245ms |
| 40000 | 246ms |
| 45000 | 242ms |
| 50000 | 242ms |

**Degree of parallelism : 16**

| cut off value | time of execution |
| --- | --- |
| 5000 | 630ms |
| 10000 | 867ms |
| 15000 | 433ms |
| 20000 | 433ms |
| 25000 | 353ms |
| 30000 | 298ms |
| 35000 | 326ms |
| 40000 | 298ms |
| 45000 | 243ms |
| 50000 | 236ms |

**Degree of parallelism : 32**

| cut off value | time of execution |
| --- | --- |
| 5000 | 663ms |
| 10000 | 939ms |
| 15000 | 388ms |
| 20000 | 320ms |
| 25000 | 341ms |
| 30000 | 390ms |
| 35000 | 408ms |
| 40000 | 409ms |
| 45000 | 358ms |
| 50000 | 320ms |

**Degree of parallelism : 64**

| cut off value | time of execution |
| --- | --- |
| 5000 | 563ms |
| 10000 | 671ms |
| 15000 | 317ms |
| 20000 | 370ms |
| 25000 | 245ms |
| 30000 | 269ms |
| 35000 | 241ms |
| 40000 | 238ms |
| 45000 | 236ms |
| 50000 | 239ms |

**For array size of 2000000, cutoff values are considered from 5000 to 50000**

**Degree of parallelism :2**

| cut off value | time of execution |
| --- | --- |
| 5000 | 1119ms |
| 10000 | 913ms |
| 15000 | 899ms |
| 20000 | 880ms |
| 25000 | 879ms |
| 30000 | 879ms |
| 35000 | 861ms |
| 40000 | 860ms |
| 45000 | 860ms |
| 50000 | 1068ms |

**Degree of parallelism :4**

| cut off value | time of execution |
| --- | --- |
| 5000 | 1018ms |
| 10000 | 719ms |
| 15000 | 640ms |
| 20000 | 628ms |
| 25000 | 631ms |
| 30000 | 617ms |
| 35000 | 603ms |
| 40000 | 608ms |
| 45000 | 604ms |
| 50000 | 607ms |

**Degree of parallelism :8**

| cut off value | time of execution |
| --- | --- |
| 5000 | 926ms |
| 10000 | 667ms |
| 15000 | 590ms |
| 20000 | 587ms |
| 25000 | 576ms |
| 30000 | 563ms |
| 35000 | 693ms |
| 40000 | 598ms |
| 45000 | 566ms |
| 50000 | 551ms |

**Degree of parallelism :16**

| cut off value | time of execution |
| --- | --- |
| 5000 | 944ms |
| 10000 | 707ms |
| 15000 | 555ms |
| 20000 | 536ms |
| 25000 | 532ms |
| 30000 | 529ms |
| 35000 | 524ms |
| 40000 | 524ms |
| 45000 | 525ms |
| 50000 | 529ms |

**Degree of parallelism :32**

| cut off value | time of execution |
| --- | --- |
| 5000 | 1082ms |
| 10000 | 721ms |
| 15000 | 554ms |
| 20000 | 502ms |
| 25000 | 488ms |
| 30000 | 502ms |
| 35000 | 505ms |
| 40000 | 497ms |
| 45000 | 501ms |
| 50000 | 501ms |

**Degree of parallelism :64**

| cut off value | time of execution |
| --- | --- |
| 5000 | 1097ms |
| 10000 | 796ms |
| 15000 | 560ms |
| 20000 | 541ms |
| 25000 | 532ms |
| 30000 | 537ms |
| 35000 | 521ms |
| 40000 | 521ms |
| 45000 | 522ms |
| 50000 | 521ms |

**For array size of 500000, cutoff values are considered from 5000 to 50000**

**Degree of parallelism :2**

| cut off value | time of execution |
| --- | --- |
| 5000 | 284ms |
| 10000 | 232ms |
| 15000 | 210ms |
| 20000 | 204ms |
| 25000 | 190ms |
| 30000 | 193ms |
| 35000 | 184ms |
| 40000 | 182ms |
| 45000 | 184ms |
| 50000 | 184ms |

**Degree of parallelism :4**

| cut off value | time of execution |
| --- | --- |
| 5000 | 286ms |
| 10000 | 222ms |
| 15000 | 138ms |
| 20000 | 151ms |
| 25000 | 138ms |
| 30000 | 133ms |
| 35000 | 133ms |
| 40000 | 130ms |
| 45000 | 130ms |
| 50000 | 129ms |

**Degree of parallelism :8**

| cut off value | time of execution |
| --- | --- |
| 5000 | 290ms |
| 10000 | 221ms |
| 15000 | 178ms |
| 20000 | 138ms |
| 25000 | 127ms |
| 30000 | 121ms |
| 35000 | 121ms |
| 40000 | 119ms |
| 45000 | 122ms |
| 50000 | 118ms |

**Degree of parallelism :16**

| cut off value | time of execution |
| --- | --- |
| 5000 | 298ms |
| 10000 | 227ms |
| 15000 | 139ms |
| 20000 | 244ms |
| 25000 | 118ms |
| 30000 | 118ms |
| 35000 | 115ms |
| 40000 | 118ms |
| 45000 | 116ms |
| 50000 | 118ms |

**Degree of parallelism :32**

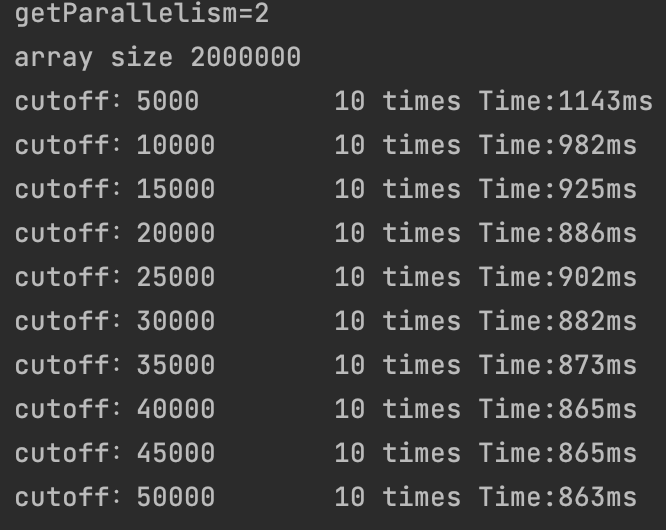
| cut off value | time of execution |
| --- | --- |
| 5000 | 458ms |
| 10000 | 246ms |
| 15000 | 160ms |
| 20000 | 160ms |
| 25000 | 142ms |
| 30000 | 116ms |
| 35000 | 117ms |
| 40000 | 116ms |
| 45000 | 117ms |
| 50000 | 117ms |

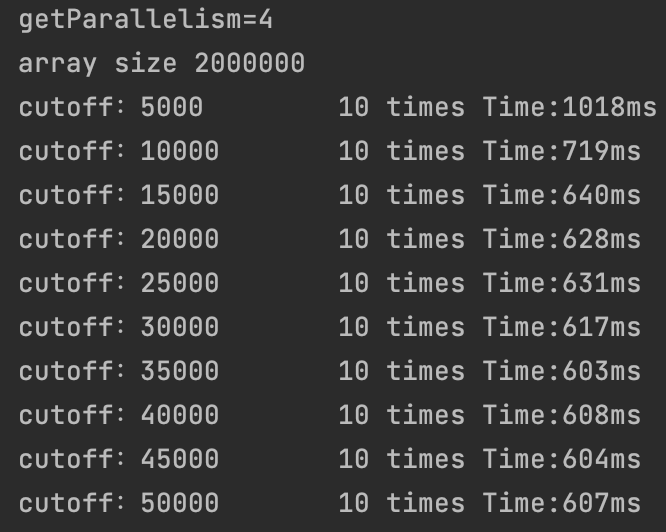
**Degree of parallelism :64**

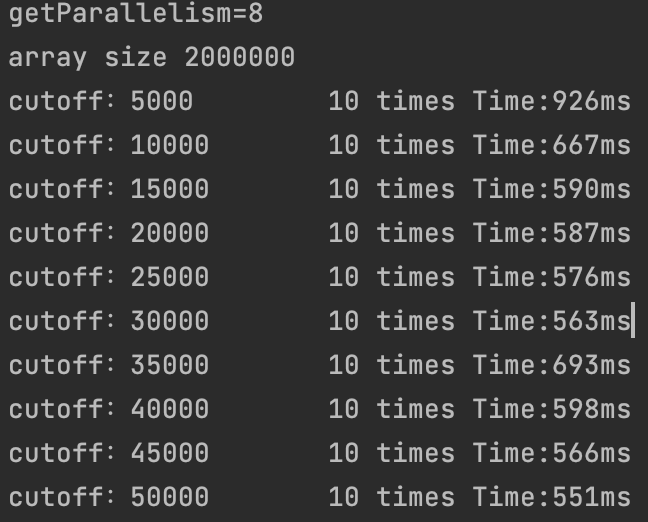
| cut off value | time of execution |
| --- | --- |
| 5000 | 355ms |
| 10000 | 275ms |
| 15000 | 127ms |
| 20000 | 125ms |
| 25000 | 124ms |
| 30000 | 122ms |
| 35000 | 119ms |
| 40000 | 117ms |
| 45000 | 116ms |
| 50000 | 117ms |

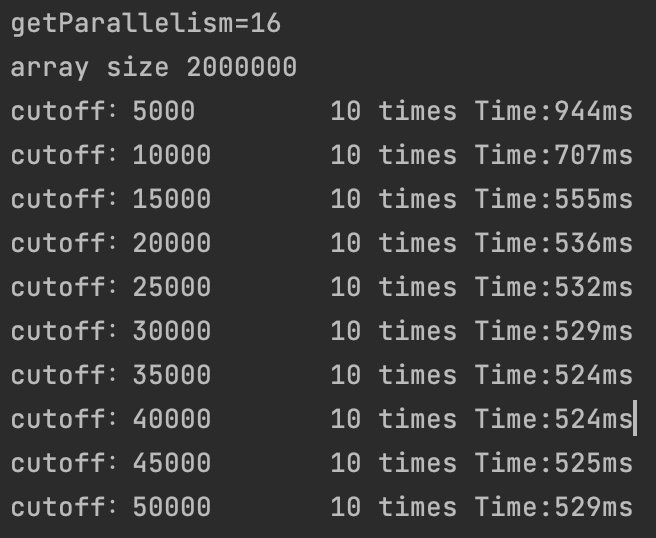
**Code Screenshots:**

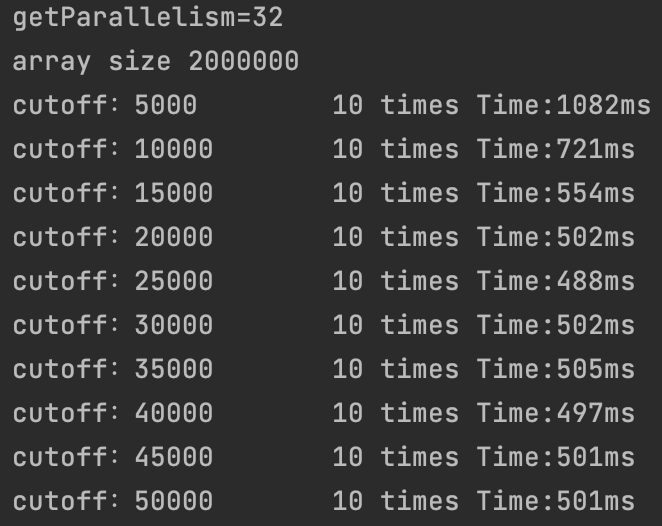
**For array size 200000**

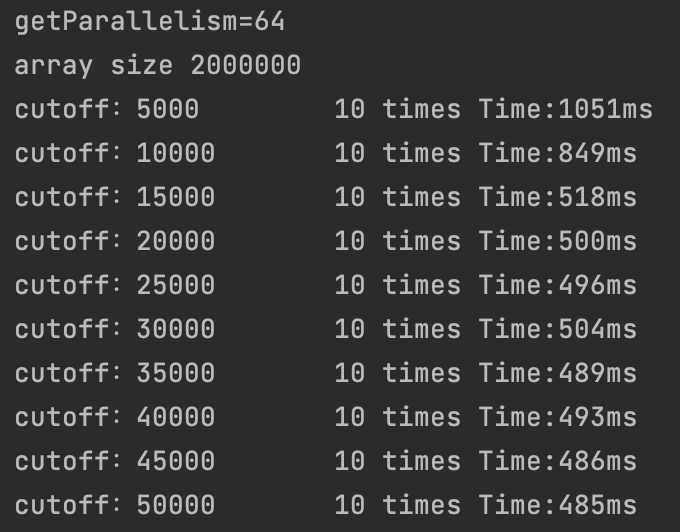




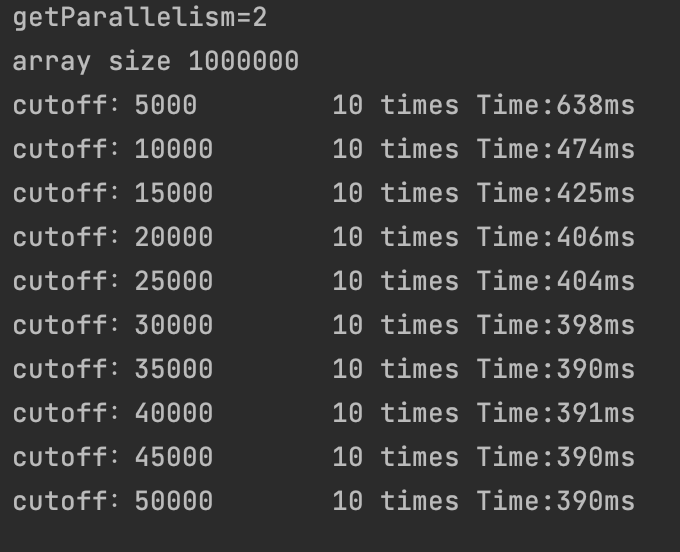


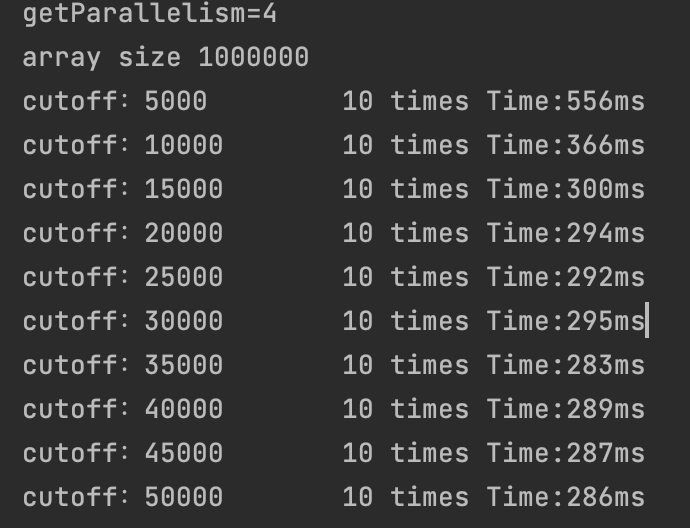


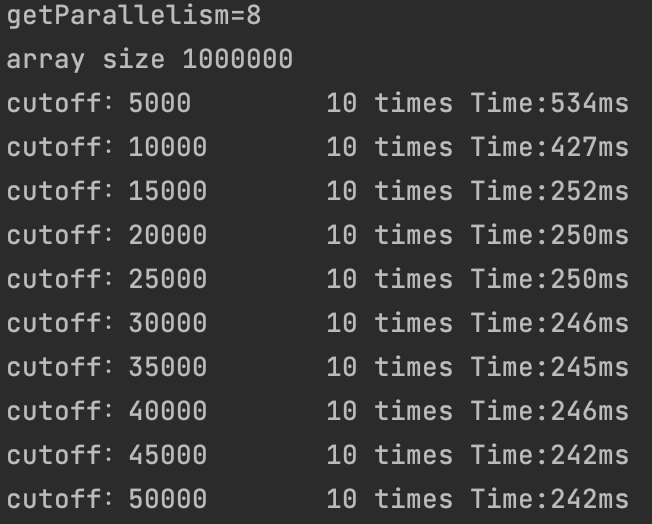


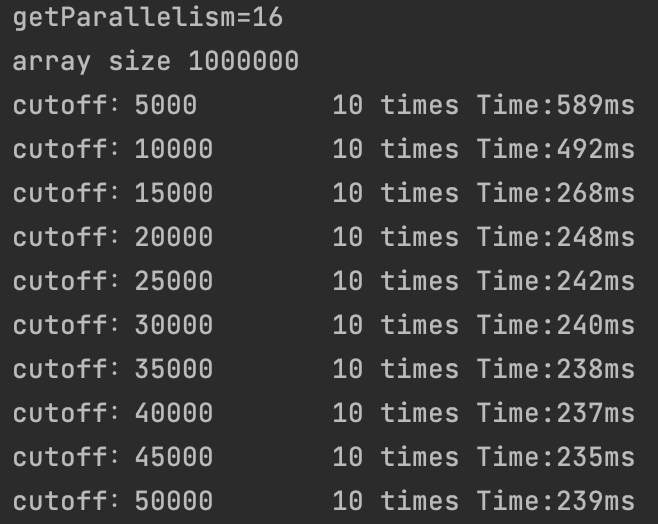


**For array size 1000000**

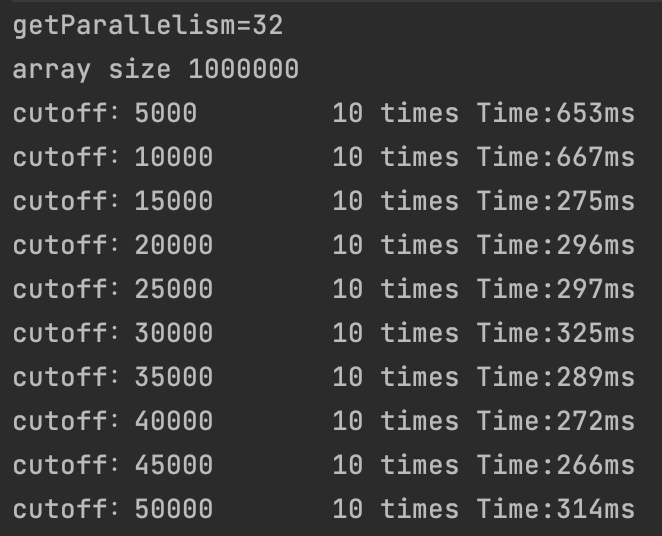


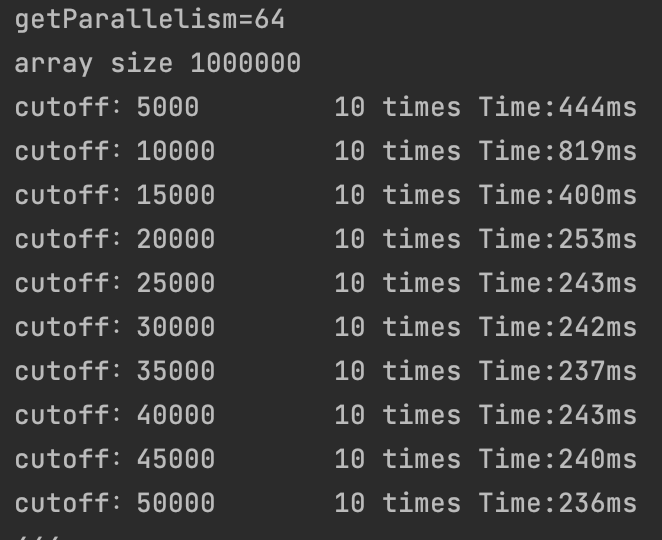






**For array size 500000**





**For array size 500000**

