# Parvataneni Mohith lalita Kumar

# (NUID : 001565574)

# INFO 6205 - Fall 2021

# Assignment No : 5

### Task:

To implement a parallel sorting algorithm such that each partition of the array is sorted in parallel

### Observations:

When array size = 2000

| Cutoff Value | Average Time for 10 runs(in milliseconds) |
|---|---|
| 100 | 55 |
| 200 | 29 |
| 250 | 18 |
| 325 | 14 |
| 500 | 11 |
| 600 | 11 |
| 700 | 9 |
| 800 | 3 |
| 900 | 3 |
| 1000 | 2 |
| 2000 | 1 |

When array size = 4000

| Cutoff Value | Average Time for 10 runs(in milliseconds) |
|---|---|
| 100 | 157 |
| 200 | 51 |
| 250 | 27 |
| 325 | 15 |
| 500 | 13 |
| 600 | 11 |
| 700 | 11 |
| 800 | 6 |
| 900 | 5 |
| 1000 | 5 |
| 2000 | 3 |

When array size = 8000

| Cutoff Value | Average Time for 10 runs(in milliseconds) |
|---|---|
| 100 | 244 |
| 200 | 80 |
| 250 | 34 |
| 325 | 26 |
| 500 | 18 |
| 600 | 17 |
| 700 | 14 |
| 800 | 13 |
| 900 | 11 |
| 1000 | 7 |
| 2000 | 5 |

When array size = 16000

| Cutoff Value | Average Time for 10 runs(in milliseconds) |
|---|---|
| 100 | 324 |
| 200 | 127 |
| 250 | 61 |
| 325 | 46 |
| 500 | 40 |
| 600 | 27 |
| 700 | 18 |
| 800 | 21 |
| 900 | 11 |
| 1000 | 9 |
| 2000 | 12 |

When array size = 32000

| Cutoff Value | Average Time for 10 runs(in milliseconds) |
|---|---|
| 100 | 331 |
| 200 | 70 |
| 250 | 58 |
| 325 | 34 |
| 500 | 25 |
| 600 | 40 |
| 700 | 15 |
| 800 | 14 |
| 900 | 19 |
| 1000 | 19 |
| 2000 | 13 |

When array size = 64000

| Cutoff Value | Average Time for 10 runs(in milliseconds) |
|---|---|
| 100 | 703 |
| 200 | 188 |
| 250 | 62 |
| 325 | 31 |
| 500 | 26 |
| 600 | 32 |
| 700 | 32 |
| 800 | 29 |
| 900 | 26 |
| 1000 | 19 |
| 2000 | 15 |

When Array size = 64000

Degree of parallelism: 2

cutoff： 200 10times Time:1179ms

cutoff： 300 10times Time:78ms

cutoff： 400 10times Time:49ms

cutoff： 500 10times Time:36ms

cutoff： 600 10times Time:31ms

cutoff： 700 10times Time:45ms

cutoff： 800 10times Time:37ms

cutoff： 900 10times Time:46ms

cutoff： 1000 10times Time:34ms

cutoff：1100 10times Time:21ms

cutoff：1200 10times Time:31ms

cutoff：1300 10times Time:22ms

cutoff：1400 10times Time:26ms

cutoff：1500 10times Time:29ms

cutoff：1600 10times Time:37ms

cutoff：1700 10times Time:35ms

cutoff：1800 10times Time:22ms

cutoff：1900 10times Time:19ms

cutoff：2000 10times Time:16ms

Degree of parallelism: 4

cutoff：200 10times Time:801ms

cutoff：300 10times Time:69ms

cutoff：400 10times Time:90ms

cutoff：500 10times Time:61ms

cutoff：600 10times Time:32ms

cutoff：700 10times Time:58ms

cutoff：800 10times Time:30ms

cutoff：900 10times Time:22ms

cutoff：1000 10times Time:16ms

cutoff：1100 10times Time:14ms

cutoff： 1200 10times Time:20ms

cutoff： 1300 10times Time:16ms

cutoff： 1400 10times Time:18ms

cutoff： 1500 10times Time:18ms

cutoff： 1600 10times Time:13ms

cutoff： 1700 10times Time:15ms

cutoff： 1800 10times Time:17ms

cutoff： 1900 10times Time:14ms

cutoff： 2000 10times Time:14ms

Degree of parallelism: 8

cutoff： 200 10times Time:760ms

cutoff： 300 10times Time:245ms

cutoff： 400 10times Time:214ms

cutoff： 500 10times Time:175ms

cutoff： 600 10times Time:178ms

cutoff： 700 10times Time:184ms

cutoff： 800 10times Time:150ms

cutoff： 900 10times Time:100ms

cutoff： 1000 10times Time:69ms

cutoff： 1100 10times Time:62ms

cutoff： 1200 10times Time:58ms

cutoff： 1300 10times Time:66ms

cutoff： 1400 10times Time:78ms

cutoff： 1500 10times Time:62ms

cutoff： 1600 10times Time:59ms

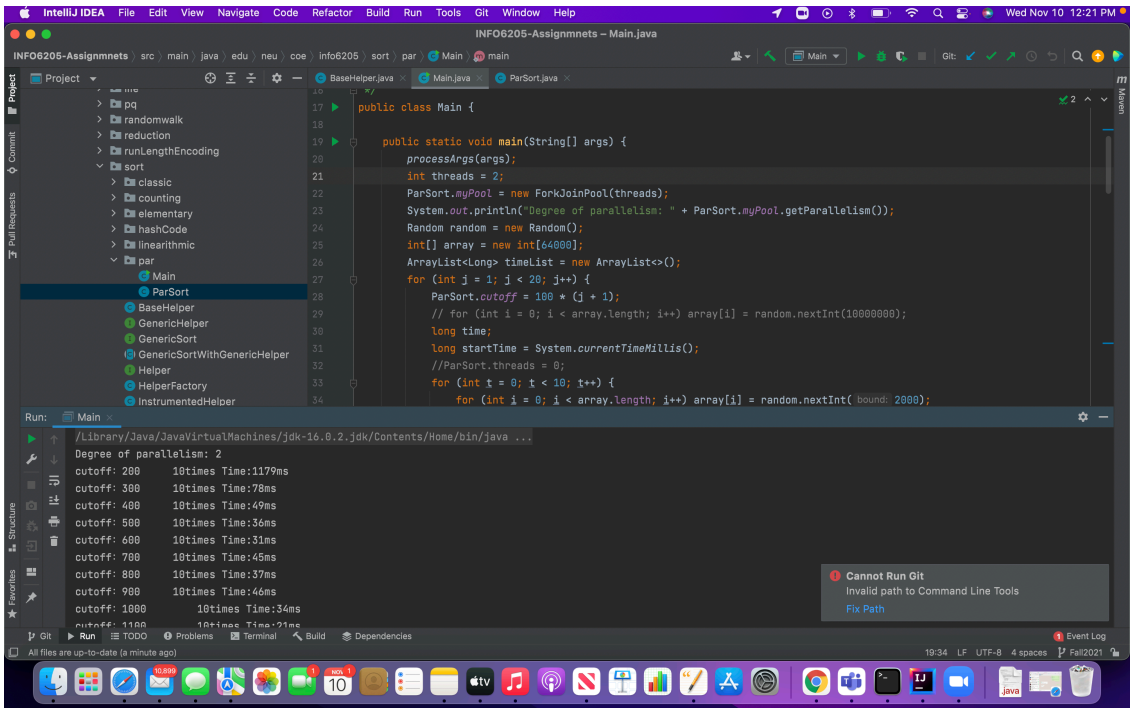
cutoff： 1700 10times Time:54ms

cutoff： 1800 10times Time:60ms

cutoff： 1900 10times Time:65ms

cutoff： 2000 10times Time:44ms

## Output Screenshot:



## Conclusion:

- For lower cutoff values, system sort is more efficient than the parallel sort
- For small array sizes, sorting becomes efficient as we increase cut off
- For larger arrays, In case of a parallel sort, based on multiple runs for different sized arrays, performance is bad for cutoff values less than 300, although, the values between 500 and 1000 see good performance results, the minimum cut off value is however 325.
- Multithreading helps in reducing the overall execution time for array provided the size of an array should be large  and cutoff is small, but when the cutoff execution time increases(thread count does not matter after certain point)

When cutoff = 325

| Size of the array | Average Time for 10 runs(in milliseconds) |
|---|---|
| 2000 | 14 |
| 4000 | 15 |
| 8000 | 26 |
| 16000 | 46 |
| 32000 | 34 |
| 64000 | 31 |