Vedant Jain (NUID: 001305281)

INFO 6205

Program Structures & Algorithms

Fall 2020

Assignment No.5

- **Task**

  You must prepare a report that shows the results of your experiments and draws a conclusion (or more) about the efficacy of this method of parallelizing sort. Your experiments should involve sorting arrays of sufficient size for the parallel sort to make a difference. You should run with many different array sizes (they must be sufficiently large to make parallel sorting worthwhile, obviously) and different cutoff schemes.

- **Output** (few outputs to prove relationship):

```
Degree of parallelism: 2
cutoff : 10000          10times Time:5378ms
cutoff : 20000          10times Time:4477ms
cutoff : 30000          10times Time:4786ms
cutoff : 40000          10times Time:4997ms
cutoff : 50000          10times Time:4763ms
cutoff : 60000          10times Time:4918ms
cutoff : 70000          10times Time:4491ms
cutoff : 80000          10times Time:5167ms
cutoff : 90000          10times Time:5288ms
cutoff : 100000         10times Time:5177ms

Degree of parallelism: 4
cutoff : 10000          10times Time:5447ms
cutoff : 20000          10times Time:4402ms
cutoff : 30000          10times Time:4400ms
cutoff : 40000          10times Time:4080ms
cutoff : 50000          10times Time:4293ms
cutoff : 60000          10times Time:4426ms
cutoff : 70000          10times Time:4424ms
cutoff : 80000          10times Time:4115ms
cutoff : 90000          10times Time:4790ms
cutoff : 100000         10times Time:5675ms
```

```
Degree of parallelism: 6
cutoff : 10000          10times Time:7051ms
cutoff : 20000          10times Time:5016ms
cutoff : 30000          10times Time:4557ms
cutoff : 40000          10times Time:5348ms
cutoff : 50000          10times Time:4294ms
cutoff : 60000          10times Time:4808ms
cutoff : 70000          10times Time:4808ms
cutoff : 80000          10times Time:5435ms
cutoff : 90000          10times Time:4652ms
cutoff : 100000         10times Time:4200ms

Degree of parallelism: 8
cutoff : 10000          10times Time:5055ms
cutoff : 20000          10times Time:4446ms
cutoff : 30000          10times Time:4529ms
cutoff : 40000          10times Time:4313ms
cutoff : 50000          10times Time:4387ms
cutoff : 60000          10times Time:4092ms
cutoff : 70000          10times Time:4223ms
cutoff : 80000          10times Time:4358ms
cutoff : 90000          10times Time:3997ms
cutoff : 100000         10times Time:4024ms

Degree of parallelism: 10
cutoff : 10000          10times Time:5084ms
cutoff : 20000          10times Time:4857ms
cutoff : 30000          10times Time:4735ms
cutoff : 40000          10times Time:4557ms
cutoff : 50000          10times Time:4221ms
cutoff : 60000          10times Time:4291ms
cutoff : 70000          10times Time:4165ms
cutoff : 80000          10times Time:4731ms
cutoff : 90000          10times Time:4779ms
cutoff : 100000         10times Time:5153ms
```

- **Conclusion on Relationship**

I changed both the number of thread and the number of cutoff in four different size of arrays. According to these outputs and graphs. I find these conclusions:

1. When the number of threads is small than 4, if we increase the number of threads, the performance of the parallel sort in the same array size and the same cutoff is better. However, when the number of threads is greater than 4 and we use the same array size and the same cutoff, the number of threads will not affect the performance of the parallel sort.
2. When the number of the threads is the same and the array size is the same, the number of cutoffs will dramatically influence the performance of the parallel sort. As the graph shown, for the same array size and the same number of threads, as the cutoffs increases, the performance of the parallel sort will perform better in the beginning, and then perform worse if cutoffs are really large.

- **Evidence to support relationship** (screen shot and/or graph and/or spreadsheet)

Array Size - 500000