## **PSA Parallel Sorting Assignment**

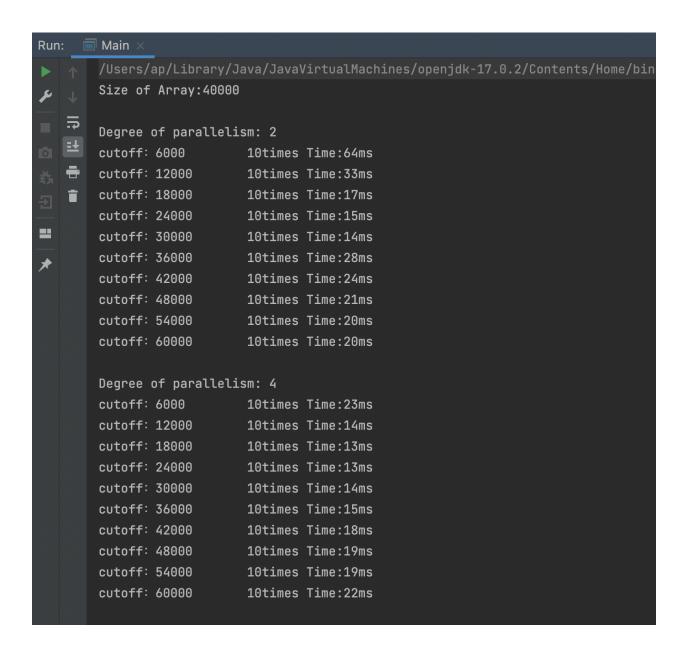
Name: Puram Anil Reddy

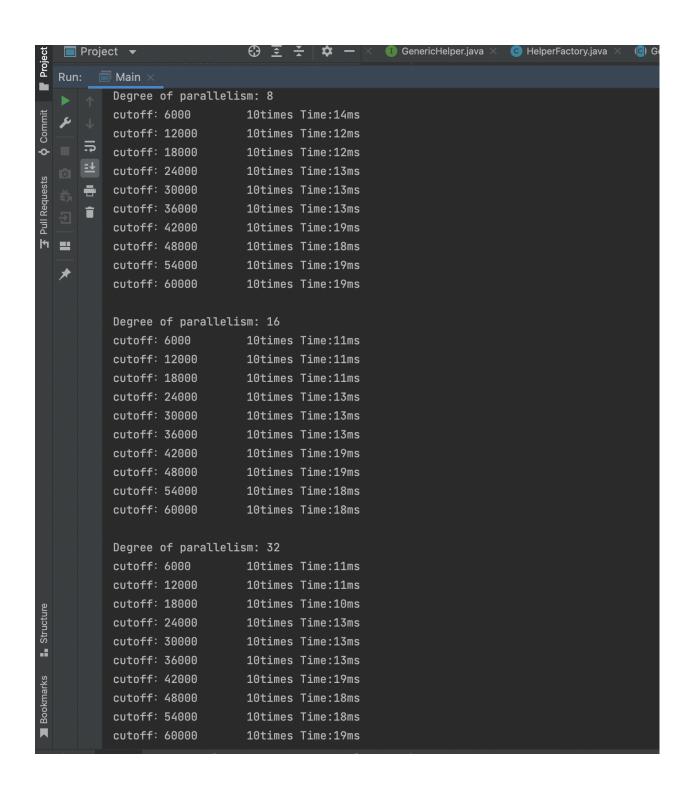
NUID:002929575

## Task:

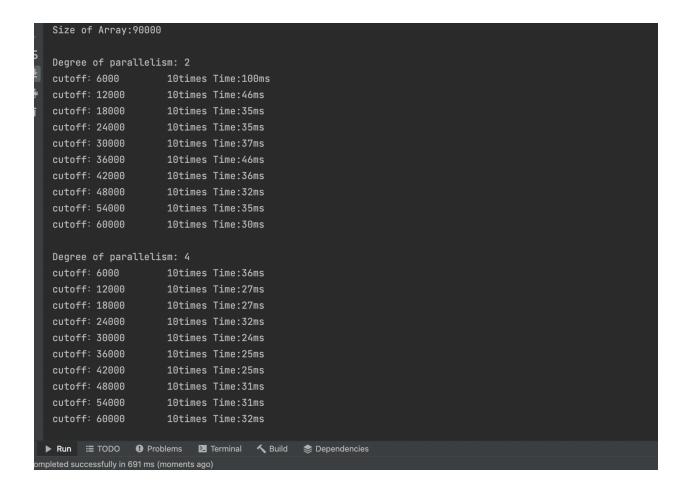
- 1. A cutoff (defaults to, say, 1000) which you will update according to the first argument in the command line when running. It's your job to experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead.
- 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.

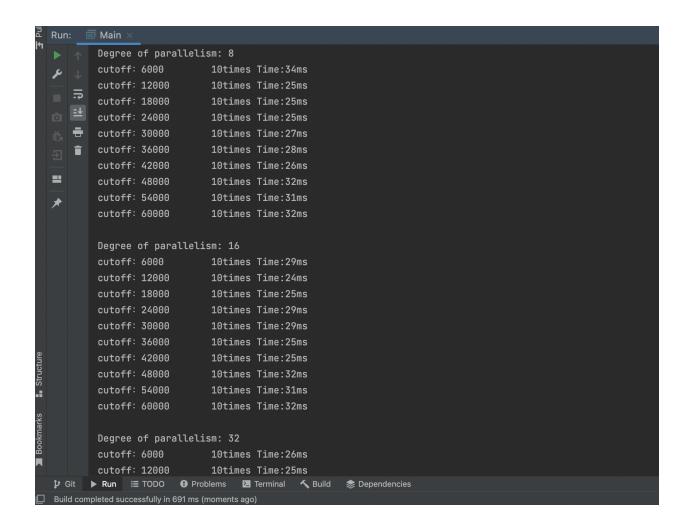
## **Output:**

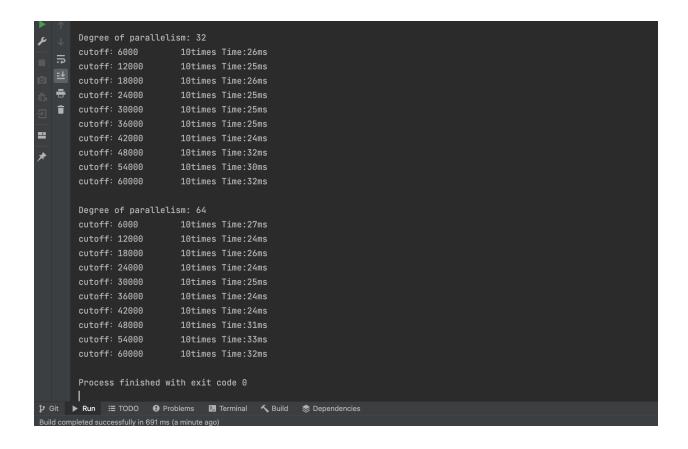




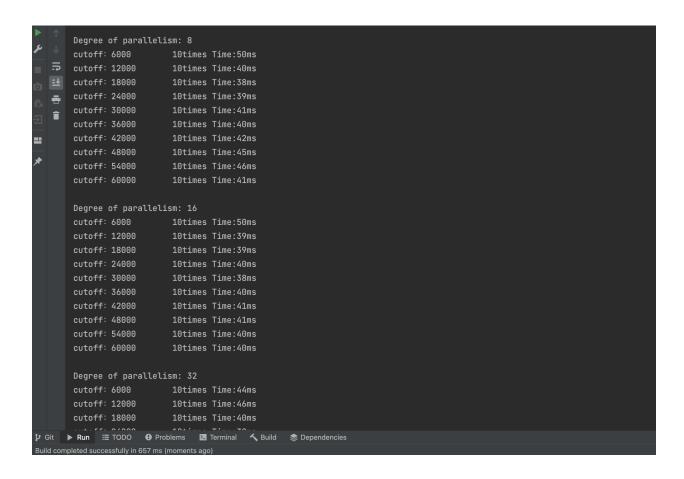


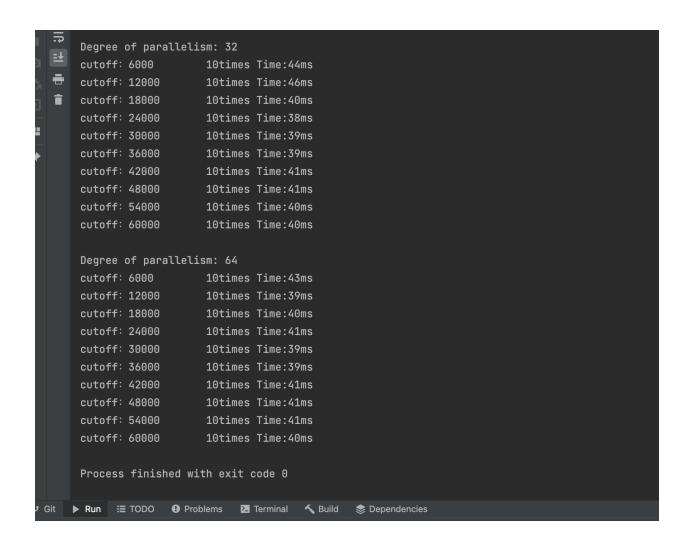






```
Size of Array:150000
    Degree of parallelism: 2
    cutoff: 6000
                        10times Time:156ms
cutoff: 12000
                        10times Time:65ms
    cutoff: 18000
                        10times Time:55ms
    cutoff: 24000
                        10times Time:77ms
    cutoff: 30000
                        10times Time:60ms
    cutoff: 36000
                        10times Time:49ms
    cutoff: 42000
                        10times Time:54ms
    cutoff: 48000
                        10times Time:54ms
    cutoff: 54000
                        10times Time:54ms
    cutoff: 60000
                        10times Time:54ms
    Degree of parallelism: 4
    cutoff: 6000
                        10times Time:54ms
    cutoff: 12000
                        10times Time:43ms
    cutoff: 18000
                        10times Time:44ms
    cutoff: 24000
                        10times Time:43ms
    cutoff: 30000
                        10times Time:44ms
    cutoff: 36000
                        10times Time:44ms
    cutoff: 42000
                        10times Time:41ms
    cutoff: 48000
                        10times Time:40ms
    cutoff: 54000
                        10times Time:41ms
    cutoff: 60000
                        10times Time:42ms
```





**Array Size: 150000** 

Cutoff	2	4	8	16	32	64
6000	438	129	181	236	346	315
12000	155	111	136	146	155	73
18000	218	123	101	66	70	54
24000	166	124	109	67	66	53
30000	157	113	82	58	67	61
36000	121	105	81	60	62	69
42000	173	107	82	61	62	64
48000	169	106	81	59	63	65
54000	99	70	70	66	67	75
60000	98	69	66	66	68	73

Array size : 90000

Cutoff	2	4	8	16	32	64
6000	307	93	53	102	143	47
12000	106	80	56	39	33	36
18000	126	64	33	30	29	29
24000	112	65	31	33	29	27
30000	46	40	30	37	32	31
36000	47	34	31	40	33	31
42000	52	34	35	37	_ 31	34
48000	47	30	Merge & Center		31	39
54000	42	41	42	42	42	91
60000	45	41	41	43	44	45

Array size : 40000

Cutoff	2	4	8	16	32	64
6000	191	52	32	50	20	18
12000	79	42	20	17	14	15
18000	90	23	16	16	15	15
24000	63	28	16	18	16	20
30000	23	27	22	21	20	24
36000	87	25	23	21	22	25
42000	59	29	26	23	23	27
48000	47	25	26	24	26	26
54000	53	35	38	35	39	36
60000	38	32	62	34	35	32

## Conclusion:

After running simulations of experiments with different combinations of the cutoff values, threads, and array sizes. Four threads are optimal beyond that there won't be much improvement

If the cutoff value is 25% of the array size, the least run time is achieved For recursion depth (d) and number of threads available (t):

t = 2 power d

Maximum depth possible:

Ig (array size lcutoff)

Any depth greater than the maximum depth is not possible because the partitioned arrays hit the cutoff and became a system sort.