Name: Namrata Ruchandani
NU ID : 002125637

Program Structures & Algorithms

Fall 2021

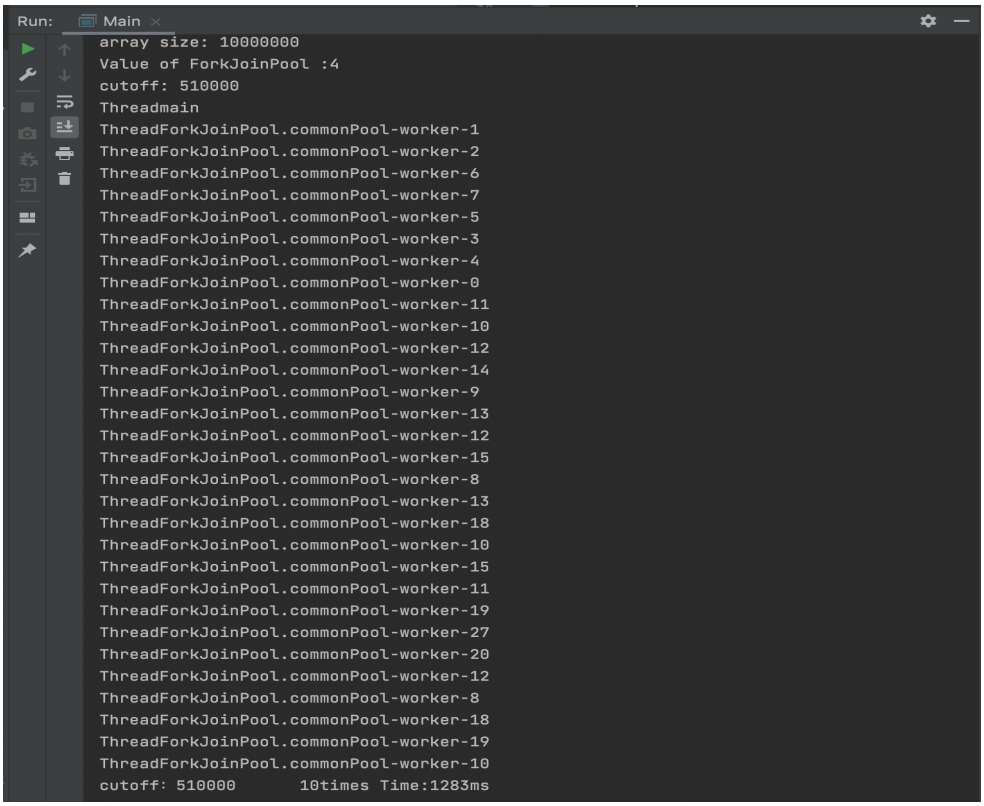
Assignment No. 5

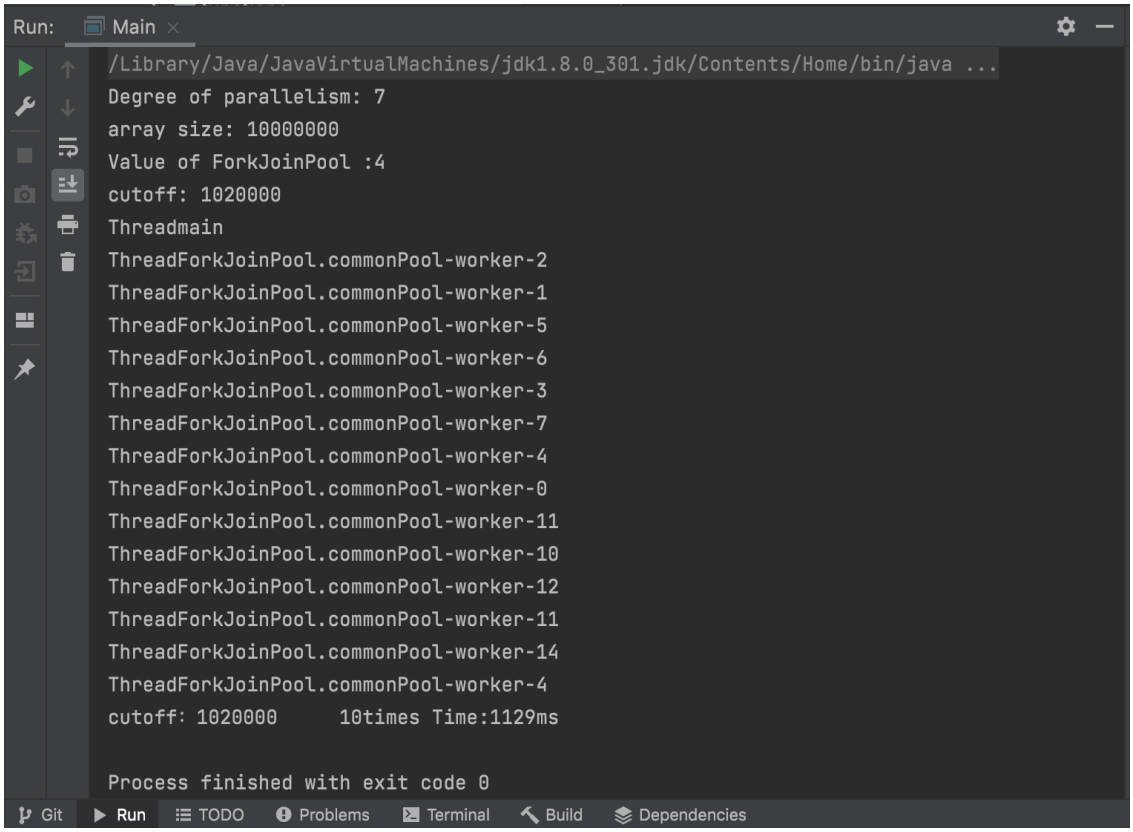
➔ Task done in this assignment:

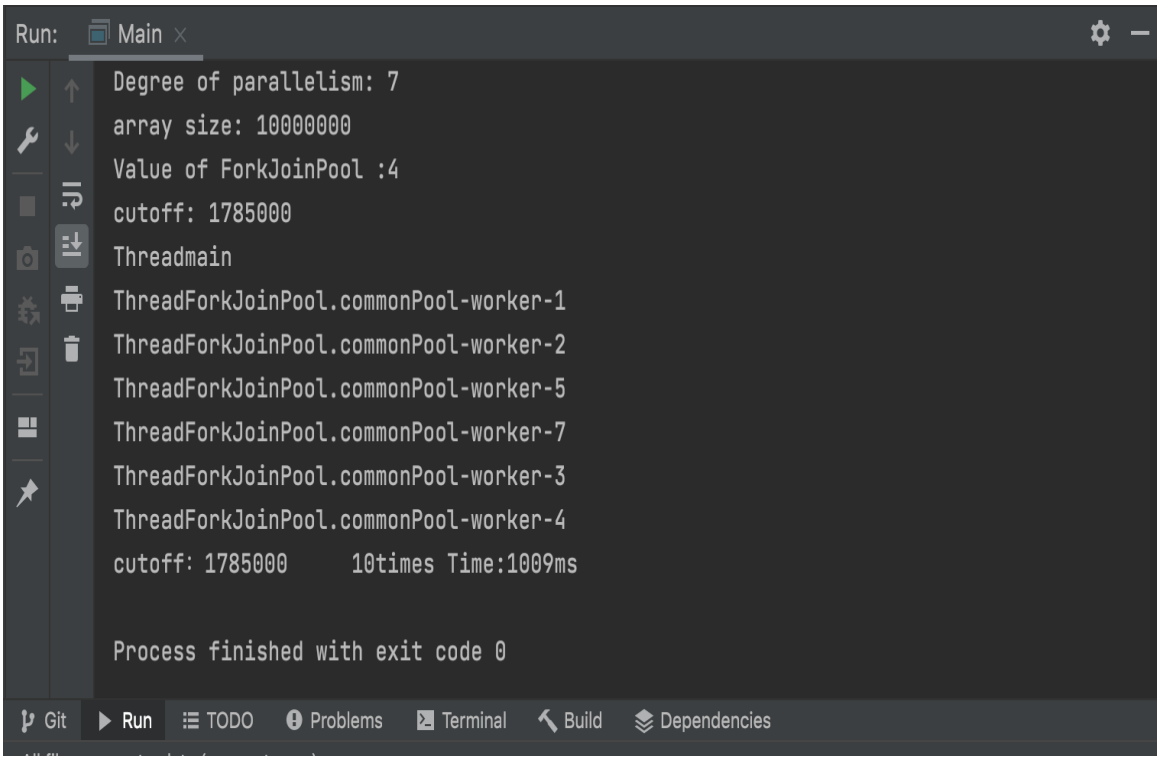
1. I executed ForkJoinPool by making an object of ForkJoinPool named as pool and passed it to parsort.
2. I then fixed my array size to 10 million, after which I compared the execution time of the function by passing different values of the number of separate threads in the power of 2,i.e., (4,8,16,64).
3. I also varied the cutoff to figure out the minimum execution time possible so that we can use the system sort instead.

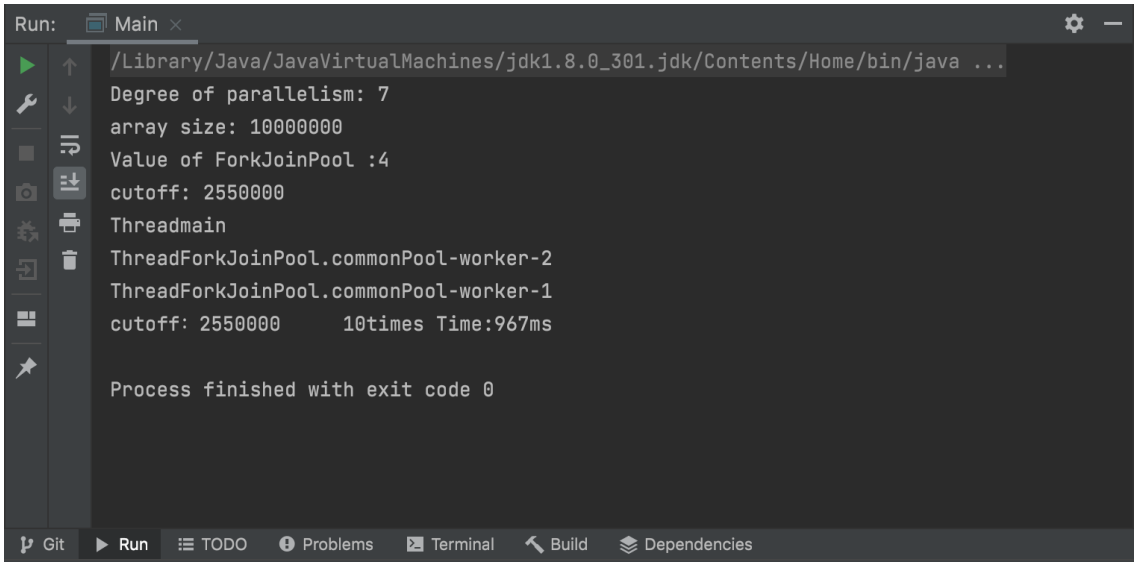
➔ The screenshot of code compilation and output

1. Number of threads - 4

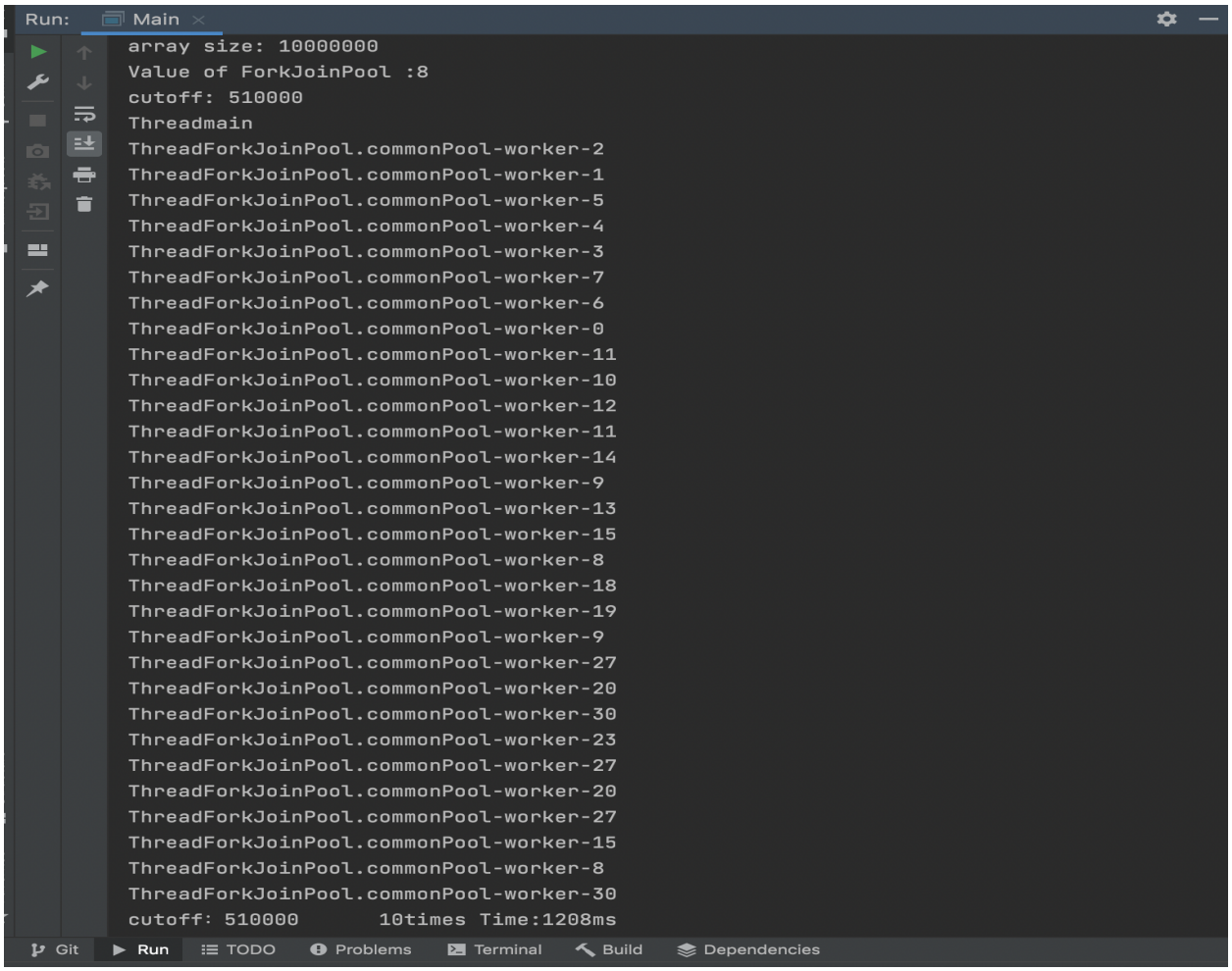


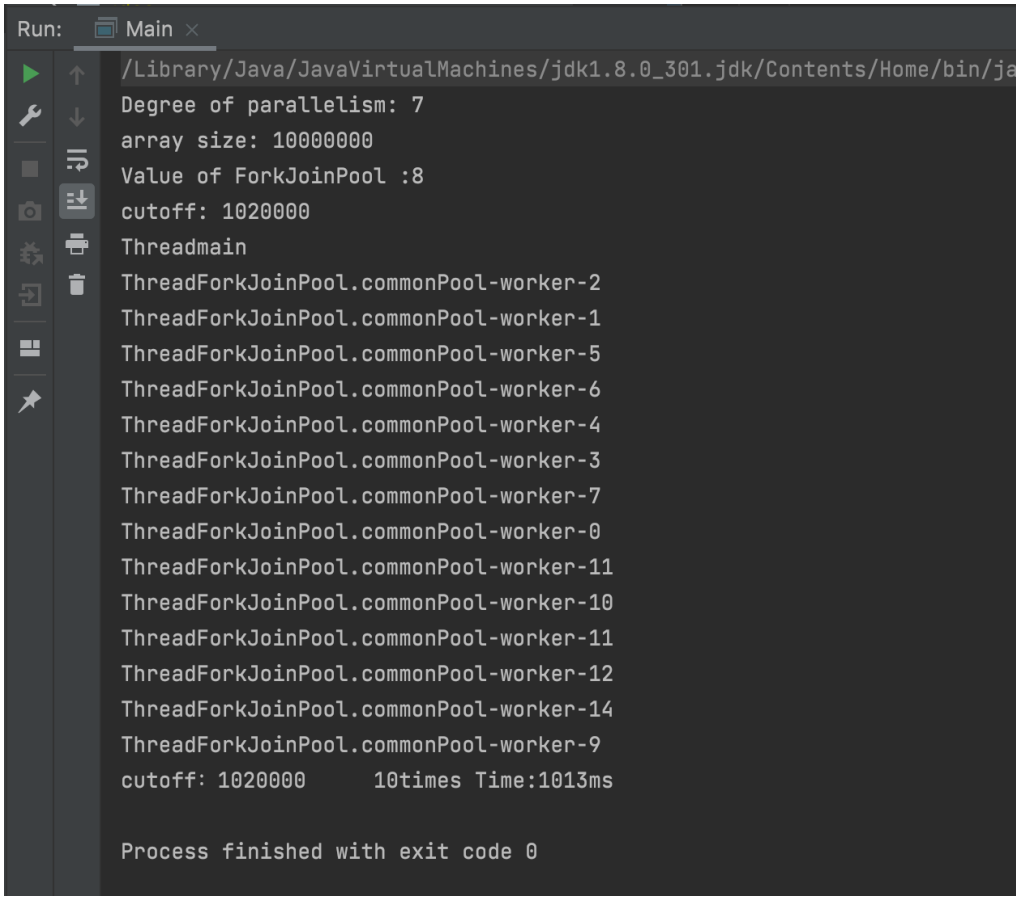

Run:
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/java ...
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :4
cutoff: 1020000
Threadmain
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-1
ThreadForkJoinPool.commonPool-worker-5
ThreadForkJoinPool.commonPool-worker-6
ThreadForkJoinPool.commonPool-worker-3
ThreadForkJoinPool.commonPool-worker-7
ThreadForkJoinPool.commonPool-worker-4
ThreadForkJoinPool.commonPool-worker-0
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-10
ThreadForkJoinPool.commonPool-worker-12
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-14
ThreadForkJoinPool.commonPool-worker-4
cutoff: 1020000 10times Time:1129ms
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Build
Dependencies

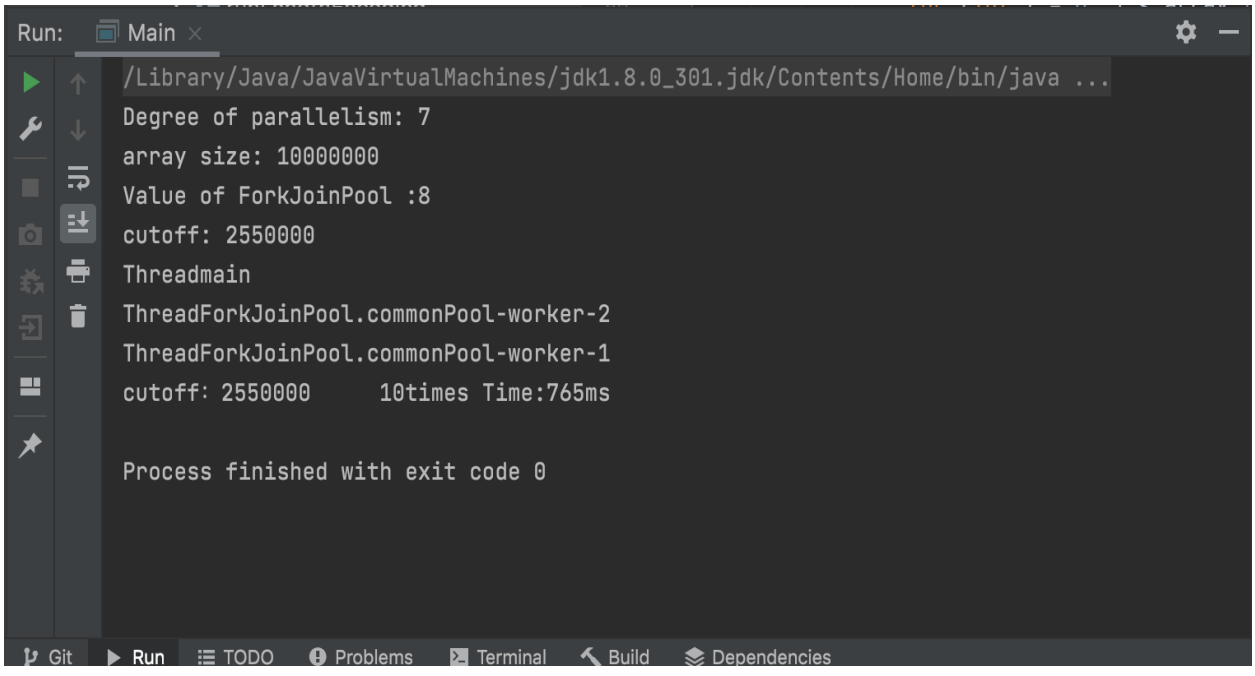

Run:
Main
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :4
cutoff: 1785000
Threadmain
ThreadForkJoinPool.commonPool-worker-1
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-5
ThreadForkJoinPool.commonPool-worker-7
ThreadForkJoinPool.commonPool-worker-3
ThreadForkJoinPool.commonPool-worker-4
cutoff: 1785000 10times Time:1009ms
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Build
Dependencies



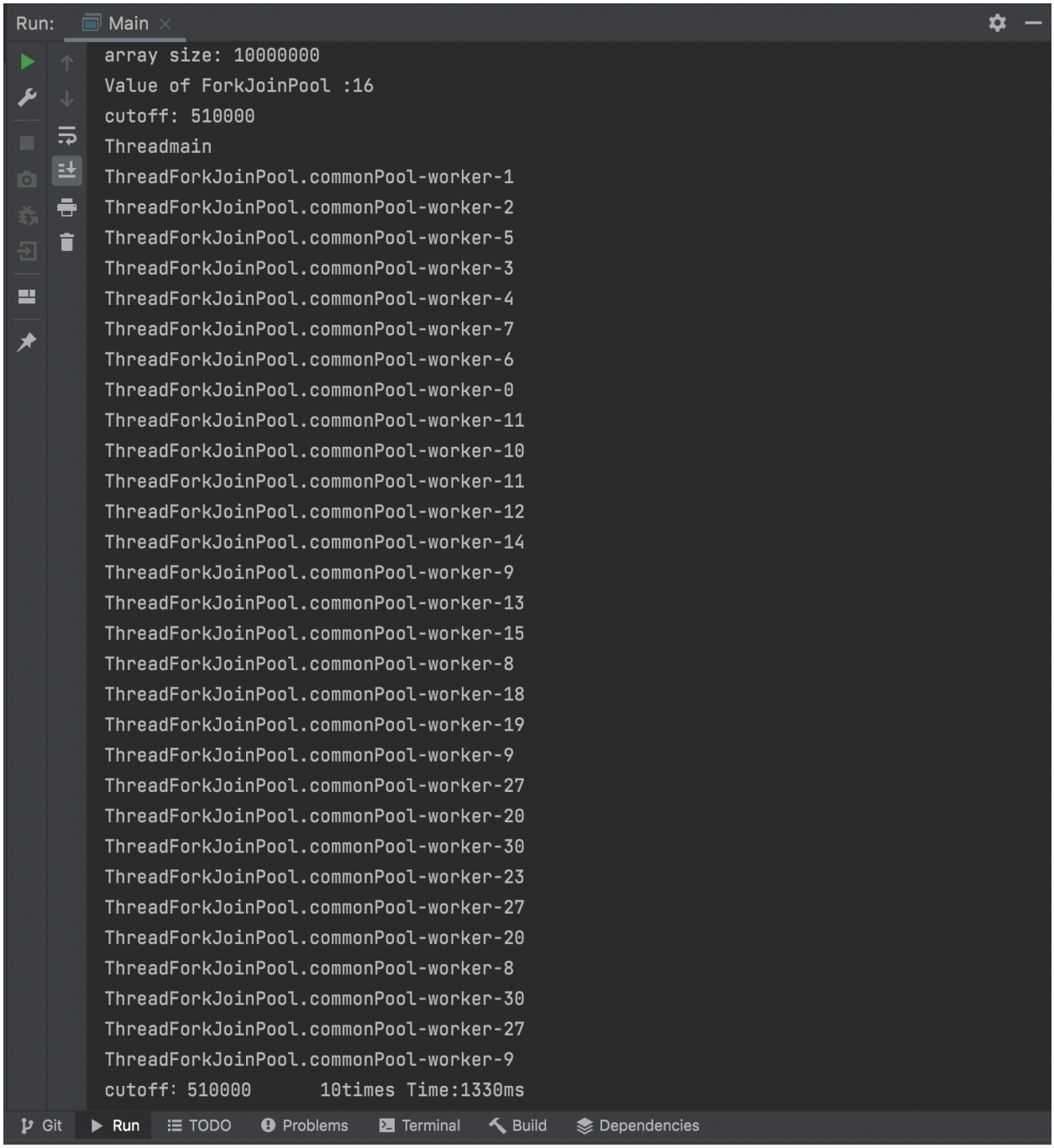
2. Number of threads - 8

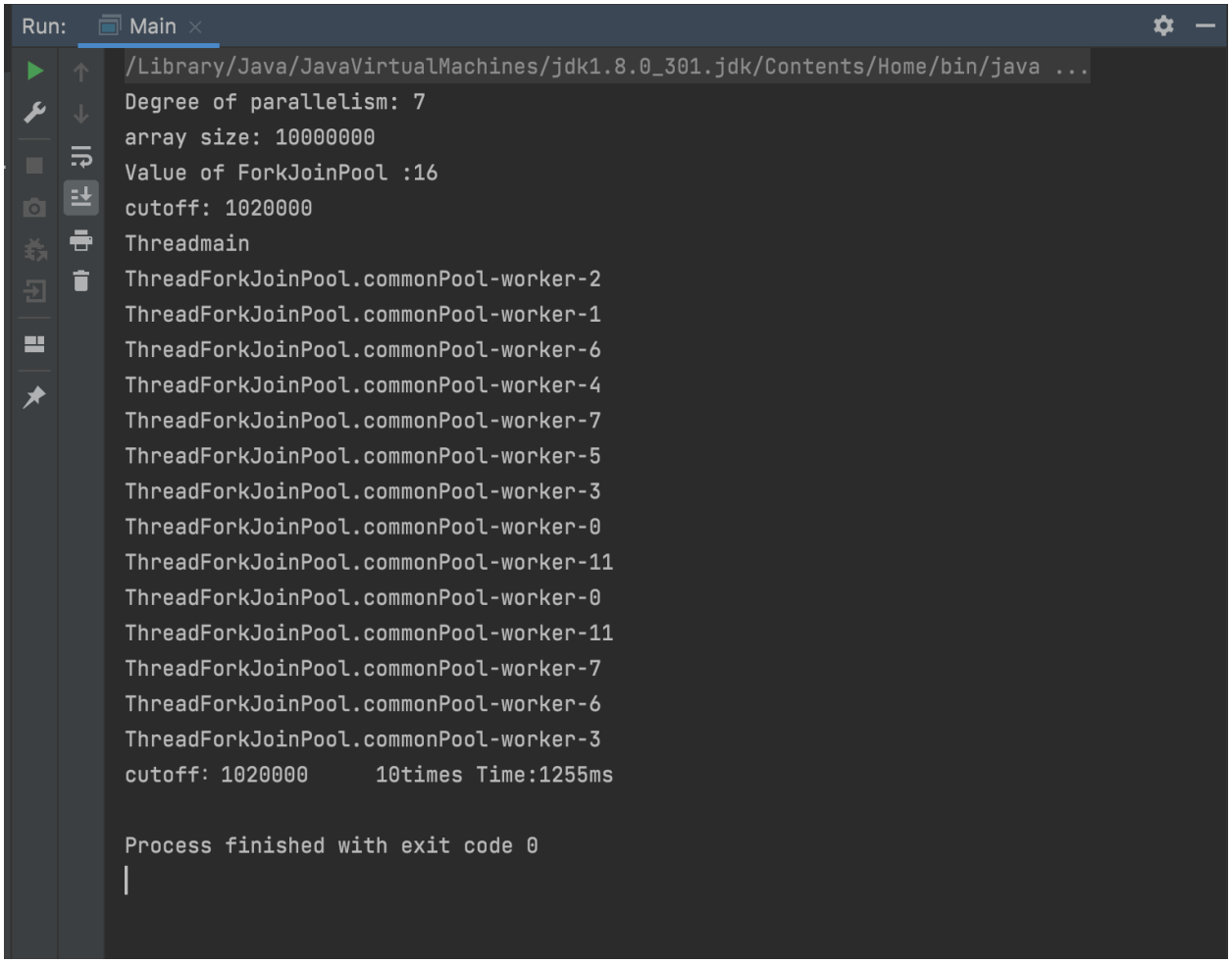


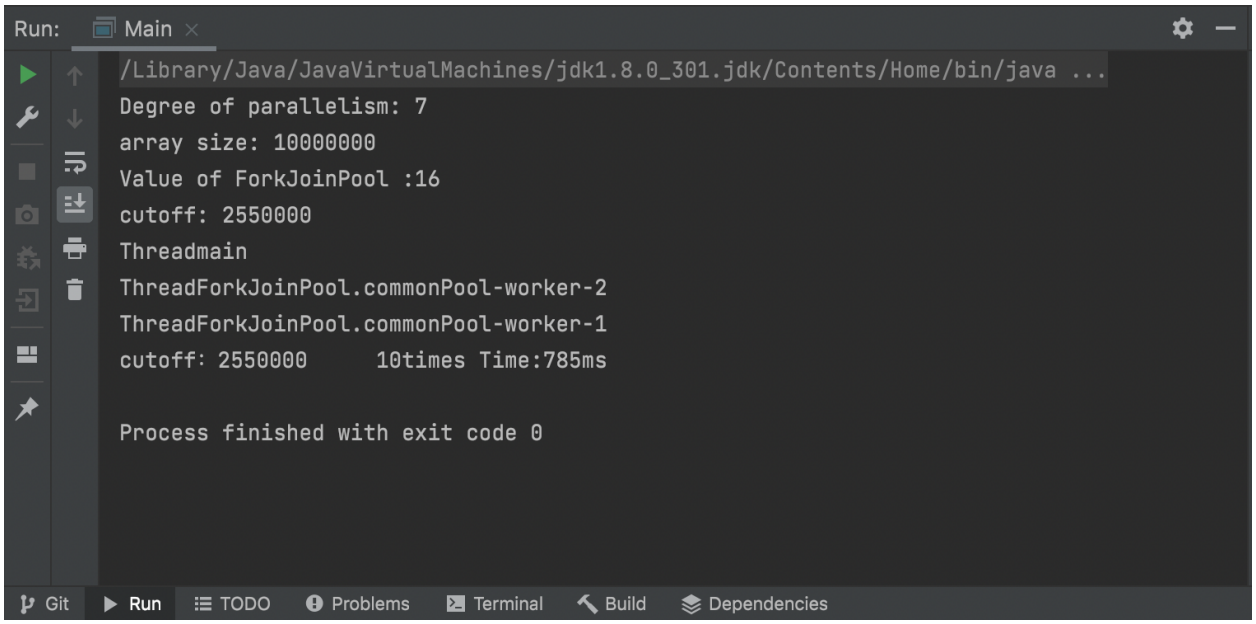

Run:
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/ja
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :8
cutoff: 1020000
Threadmain
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-1
ThreadForkJoinPool.commonPool-worker-5
ThreadForkJoinPool.commonPool-worker-6
ThreadForkJoinPool.commonPool-worker-4
ThreadForkJoinPool.commonPool-worker-3
ThreadForkJoinPool.commonPool-worker-7
ThreadForkJoinPool.commonPool-worker-0
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-10
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-12
ThreadForkJoinPool.commonPool-worker-14
ThreadForkJoinPool.commonPool-worker-9
cutoff: 1020000 10times Time:1013ms
Process finished with exit code 0


Run:
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/java ...
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :8
cutoff: 2550000
Threadmain
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-1
cutoff: 2550000 10times Time:765ms
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Build
Dependencies

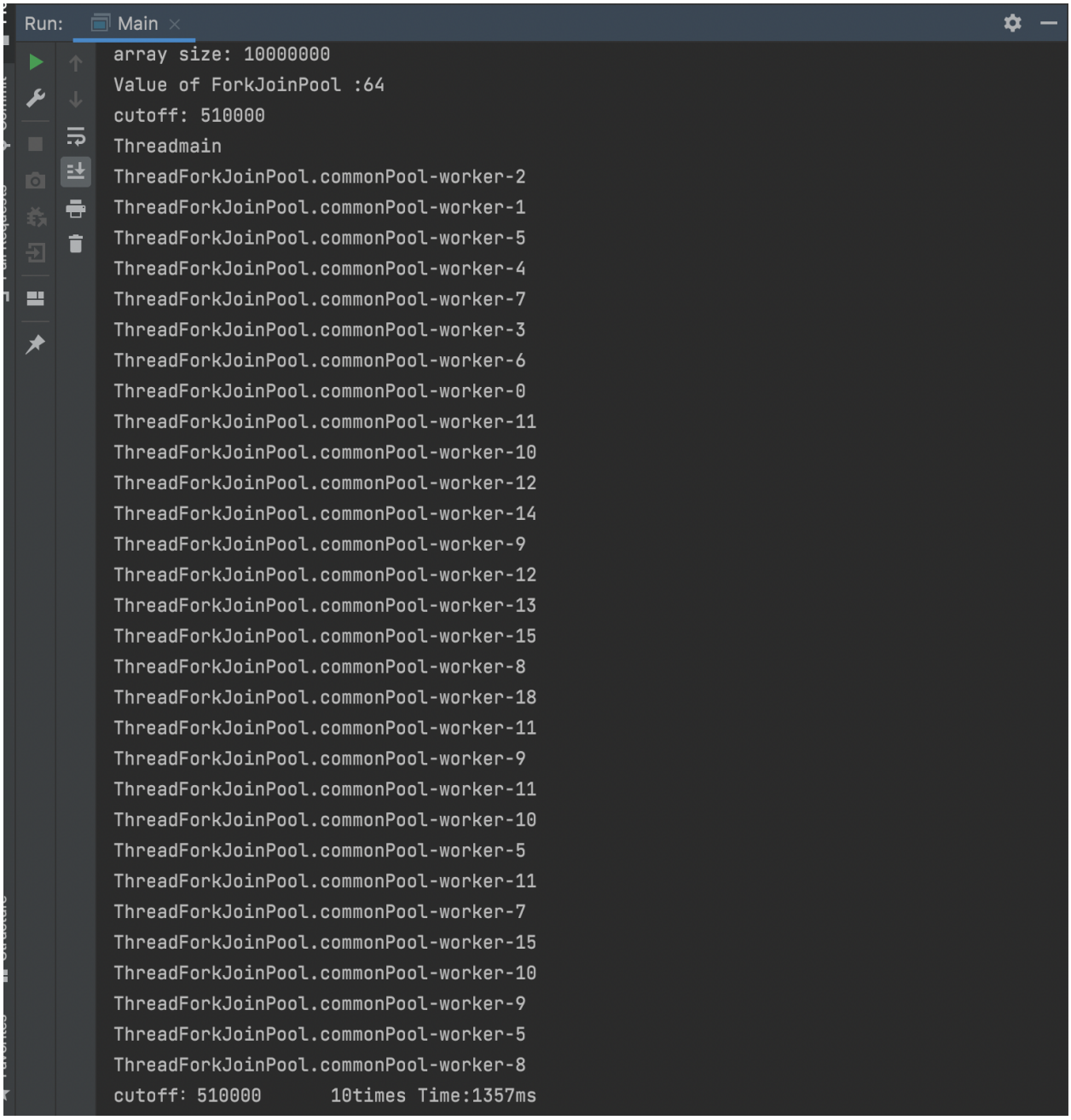
3. Number of threads - 16

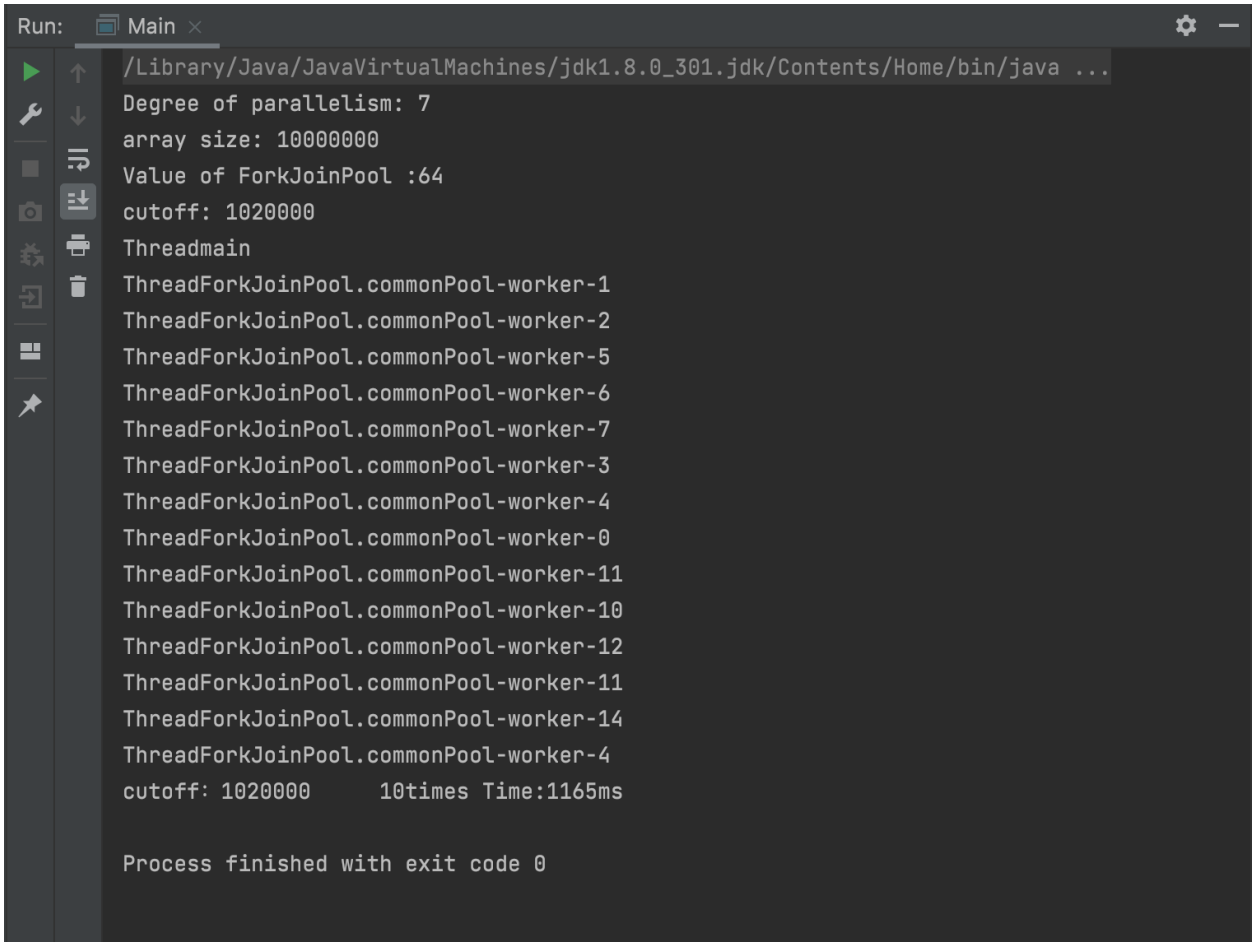


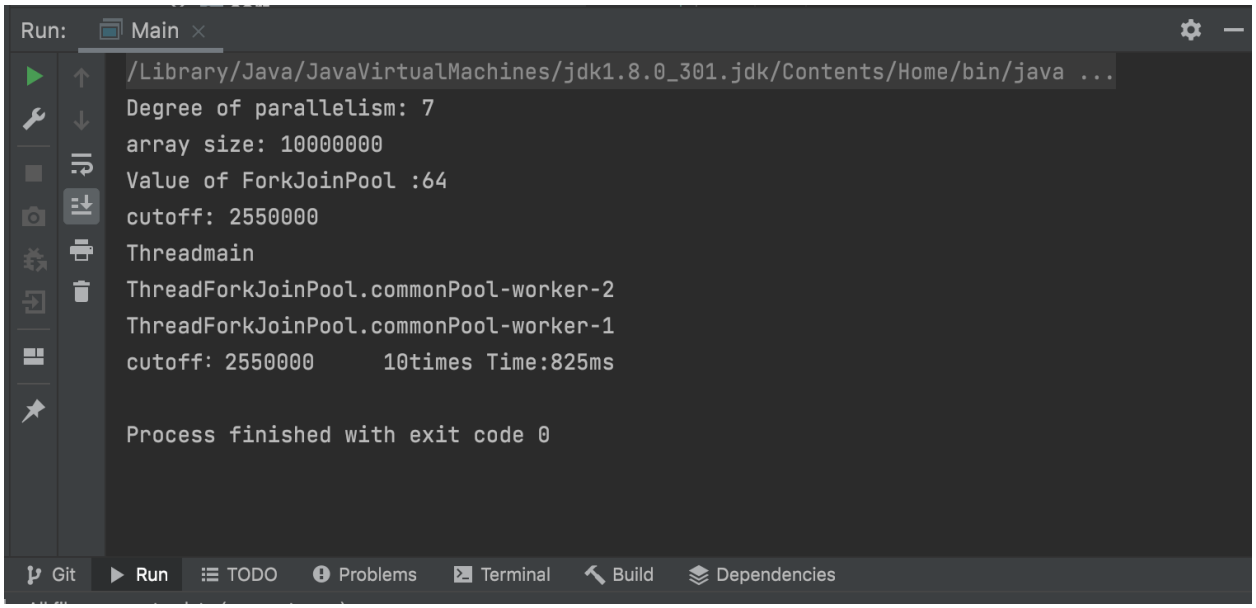

Run:
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/java ...
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :16
cutoff: 1020000
Threadmain
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-1
ThreadForkJoinPool.commonPool-worker-6
ThreadForkJoinPool.commonPool-worker-4
ThreadForkJoinPool.commonPool-worker-7
ThreadForkJoinPool.commonPool-worker-5
ThreadForkJoinPool.commonPool-worker-3
ThreadForkJoinPool.commonPool-worker-0
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-0
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-7
ThreadForkJoinPool.commonPool-worker-6
ThreadForkJoinPool.commonPool-worker-3
cutoff: 1020000 10times Time:1255ms
Process finished with exit code 0


Run:
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/java ...
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :16
cutoff: 2550000
Threadmain
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-1
cutoff: 2550000 10times Time:785ms
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Build
Dependencies

4. Number of threads - 64



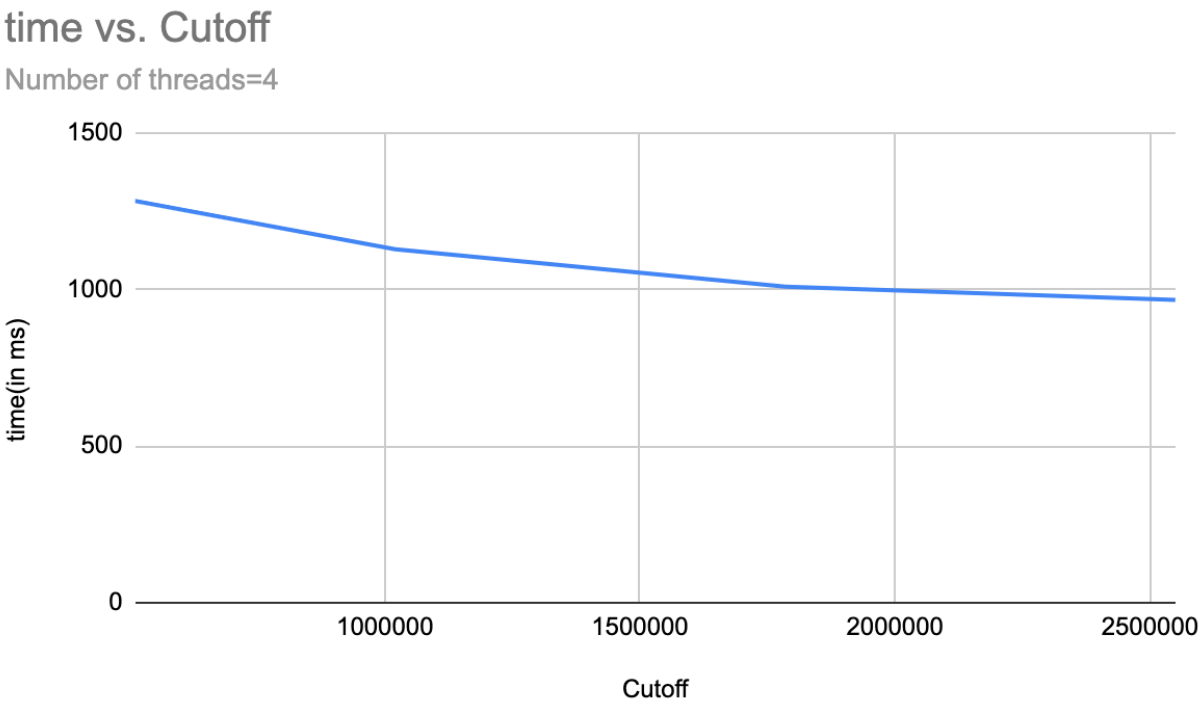

Run:
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/java ...
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :64
cutoff: 1020000
Threadmain
ThreadForkJoinPool.commonPool-worker-1
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-5
ThreadForkJoinPool.commonPool-worker-6
ThreadForkJoinPool.commonPool-worker-7
ThreadForkJoinPool.commonPool-worker-3
ThreadForkJoinPool.commonPool-worker-4
ThreadForkJoinPool.commonPool-worker-0
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-10
ThreadForkJoinPool.commonPool-worker-12
ThreadForkJoinPool.commonPool-worker-11
ThreadForkJoinPool.commonPool-worker-14
ThreadForkJoinPool.commonPool-worker-4
cutoff: 1020000 10times Time:1165ms
Process finished with exit code 0


Run:
Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Contents/Home/bin/java ...
Degree of parallelism: 7
array size: 10000000
Value of ForkJoinPool :64
cutoff: 2550000
Threadmain
ThreadForkJoinPool.commonPool-worker-2
ThreadForkJoinPool.commonPool-worker-1
cutoff: 2550000 10times Time:825ms
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Build
Dependencies

➔ **Conclusion about number of separate threads and good value of cutoff:**

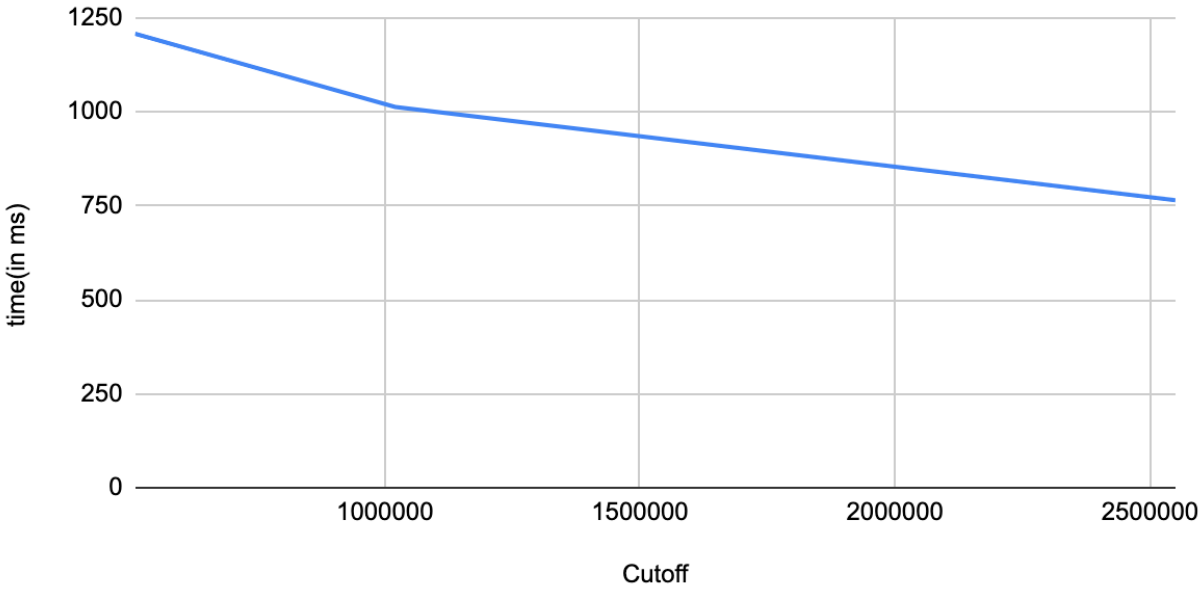
- Here, we are dividing the array into parts as separate tasks and then executing in a ForkJoinPool. This will allow us to sort parts of an array independently and then just merge them.
- I tried passing a different number of separate threads, for my system the ideal number of separate threads could be between 8 to 16.
- To decide when to sort directly, I tried passing different cutoffs. According to my experiments the good value for the cutoff is 2550000, which is a quarter of the size of the array, i.e., N/4.
- The conclusion I can draw from this assignment is Parallel Merge Sort is faster for arrays of 10M of elements or more. Although, speed is not 2x times as there are overheads for scheduling tasks in the pool, still it's a significant difference.

➔ **Below are the graphs of Time vs Cutoff for different number of threads:**



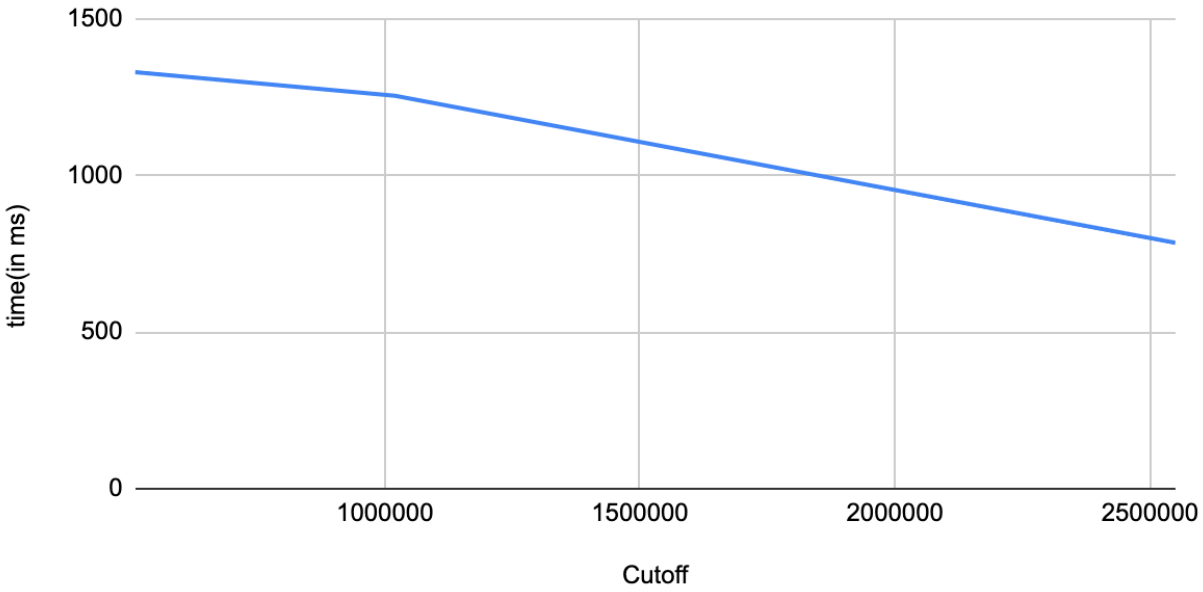
time vs. Cutoff

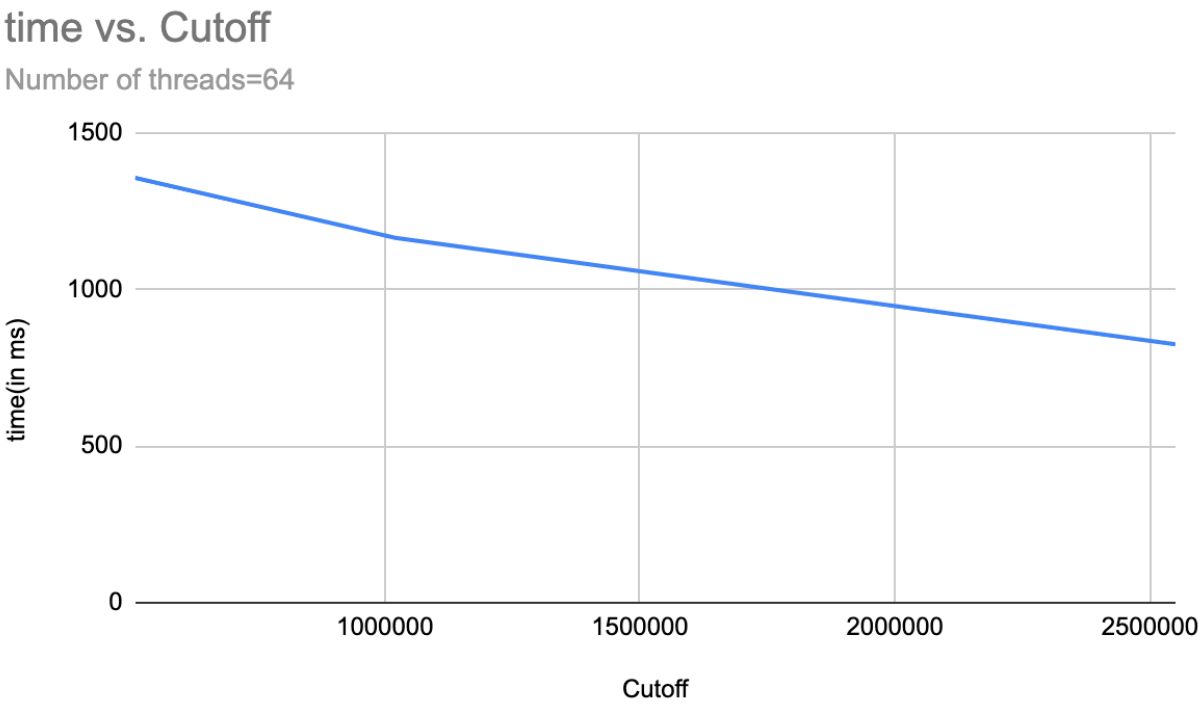
Number of threads=8



time vs. Cutoff

Number of threads=16





➔ **I pushed the commands in the git and attached the link for the repository.**
https://github.com/Namrata2108/INFO6205/tree/Fall2021/src/main/java/edu/neu/coe/info6205/sort/par