Program Structures & Algorithms

Spring 2022

Assignment No. 4

Name: Hanwen Jiang

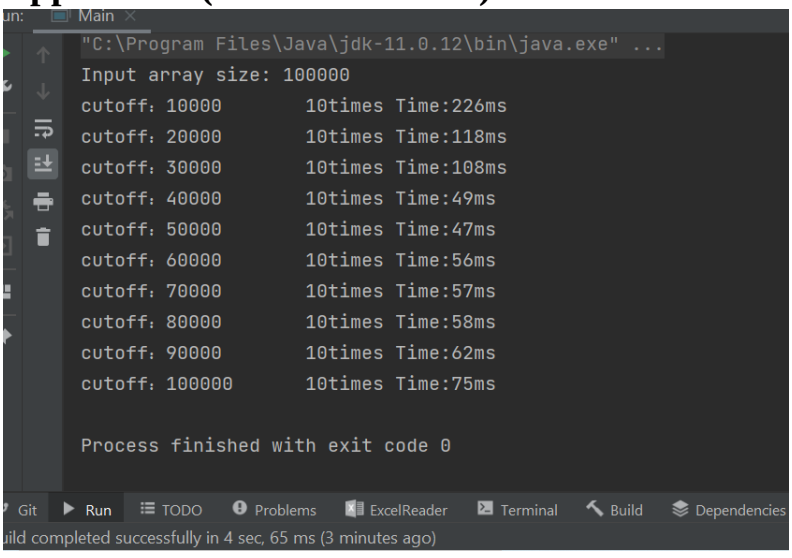
(NUID): 002199242

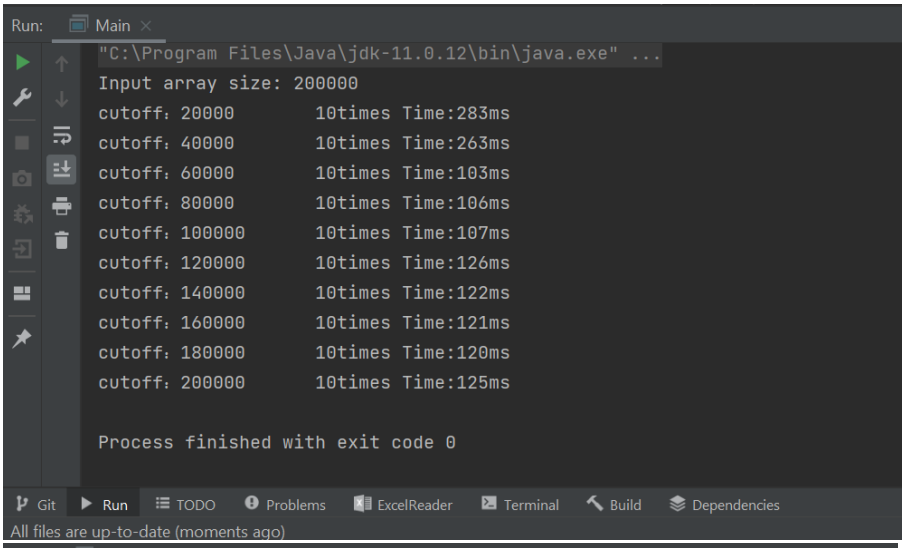
- **Task**

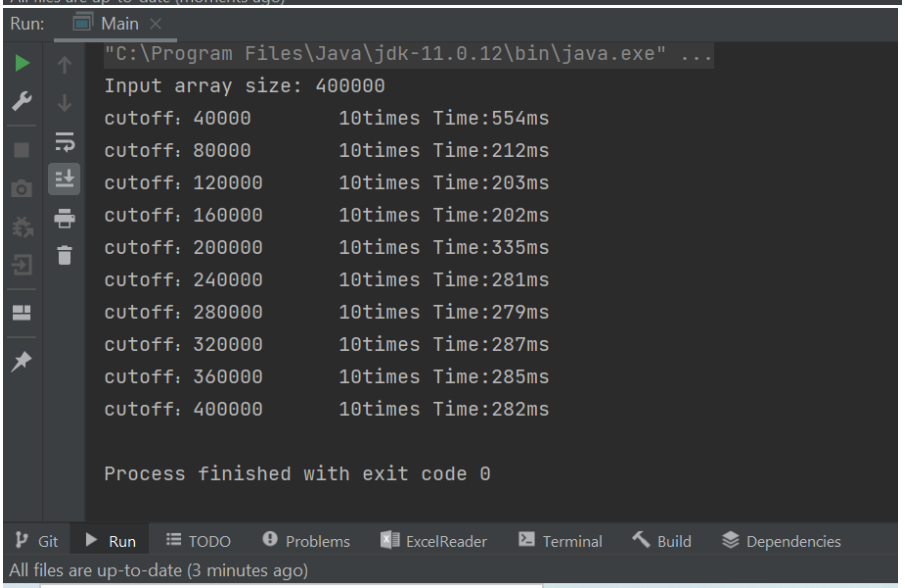
- A cutoff (defaults to, say, 1000) which you will update according to the first argument in the command line when running. It's your job to experiment and come up with a good value for this cutoff. If there are fewer elements to sort than the cutoff, then you should use the system sort instead.
- Recursion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

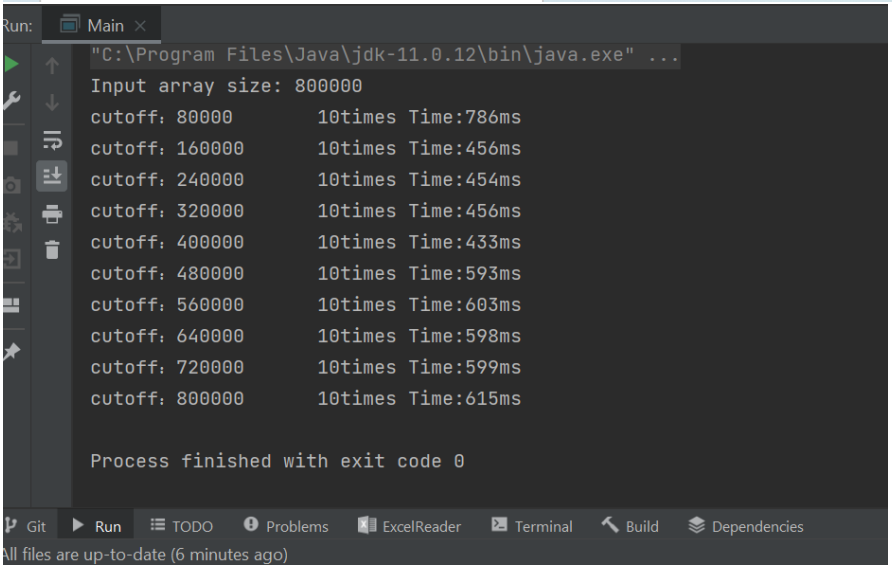
- **Output screenshot**

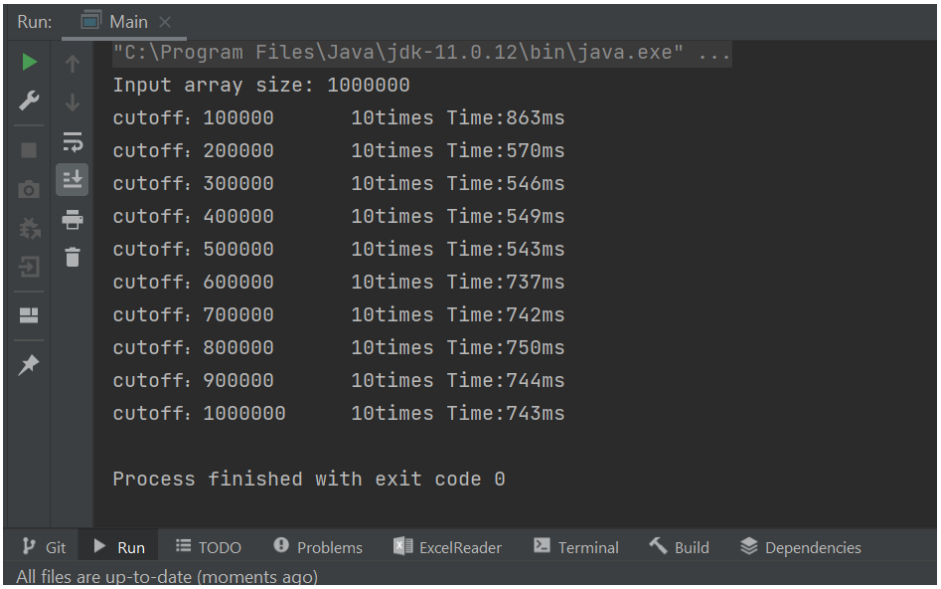
Approach 1 (based on cut offs)



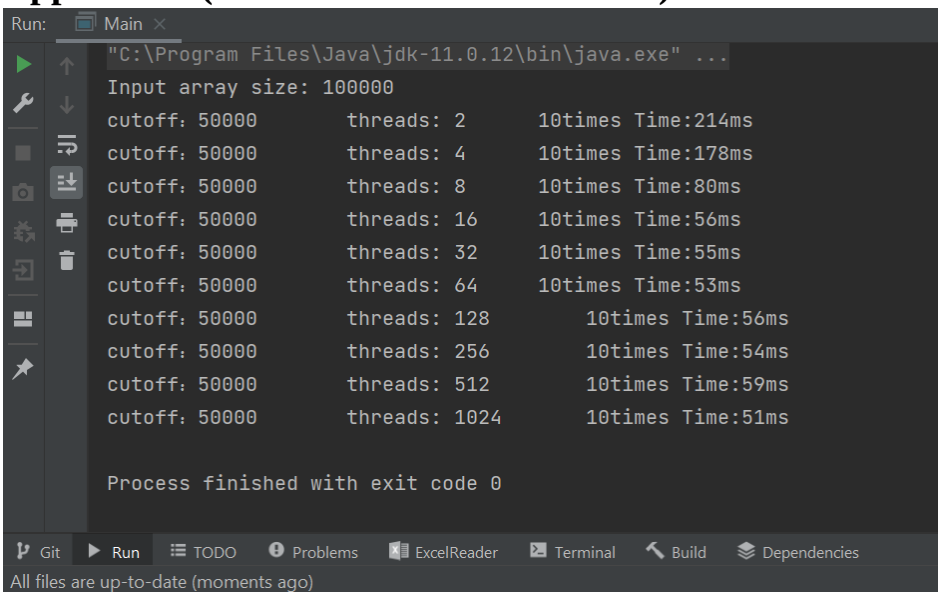

Run: Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 200000
cutoff: 20000 10times Time:283ms
cutoff: 40000 10times Time:263ms
cutoff: 60000 10times Time:103ms
cutoff: 80000 10times Time:106ms
cutoff: 100000 10times Time:107ms
cutoff: 120000 10times Time:126ms
cutoff: 140000 10times Time:122ms
cutoff: 160000 10times Time:121ms
cutoff: 180000 10times Time:120ms
cutoff: 200000 10times Time:125ms
Process finished with exit code 0
Git Run TODO Problems ExcelReader Terminal Build Dependencies
All files are up-to-date (moments ago)

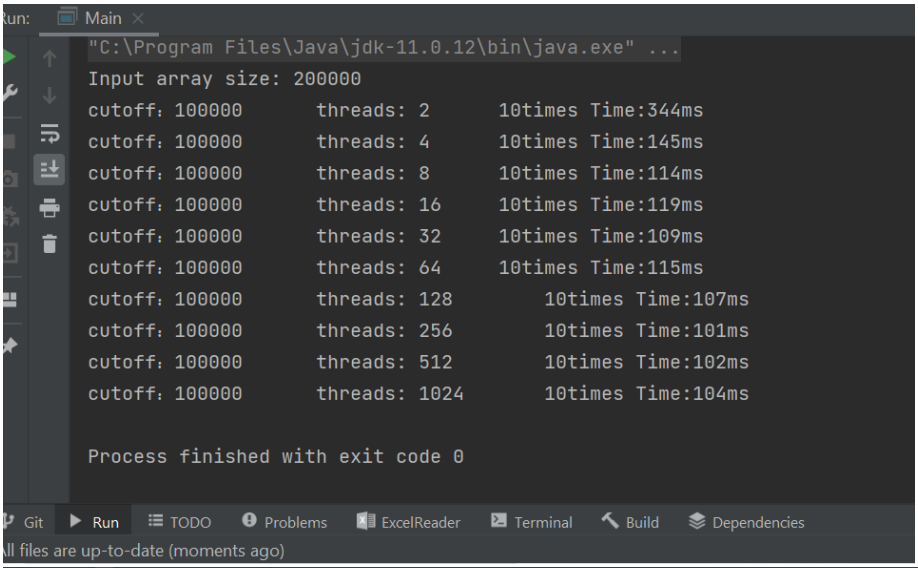

Run: Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 400000
cutoff: 40000 10times Time:554ms
cutoff: 80000 10times Time:212ms
cutoff: 120000 10times Time:203ms
cutoff: 160000 10times Time:202ms
cutoff: 200000 10times Time:335ms
cutoff: 240000 10times Time:281ms
cutoff: 280000 10times Time:279ms
cutoff: 320000 10times Time:287ms
cutoff: 360000 10times Time:285ms
cutoff: 400000 10times Time:282ms
Process finished with exit code 0
Git Run TODO Problems ExcelReader Terminal Build Dependencies
All files are up-to-date (3 minutes ago)

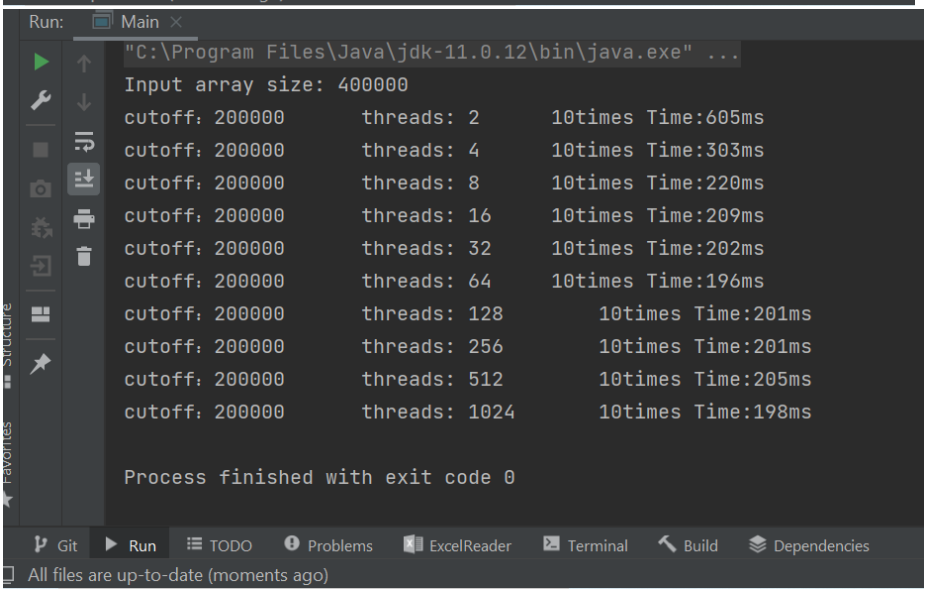

Run: Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 800000
cutoff: 80000 10times Time:786ms
cutoff: 160000 10times Time:456ms
cutoff: 240000 10times Time:454ms
cutoff: 320000 10times Time:456ms
cutoff: 400000 10times Time:433ms
cutoff: 480000 10times Time:593ms
cutoff: 560000 10times Time:603ms
cutoff: 640000 10times Time:598ms
cutoff: 720000 10times Time:599ms
cutoff: 800000 10times Time:615ms
Process finished with exit code 0
Git Run TODO Problems ExcelReader Terminal Build Dependencies
All files are up-to-date (6 minutes ago)

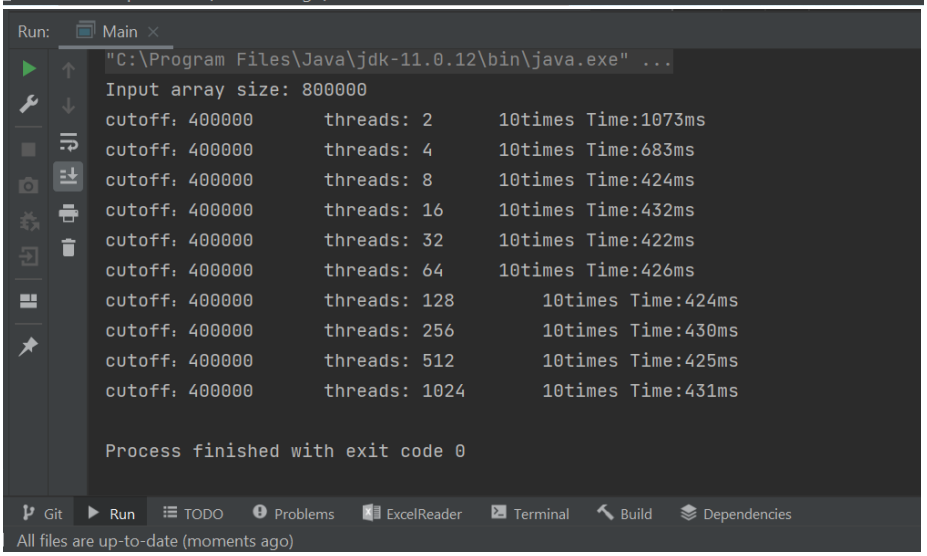


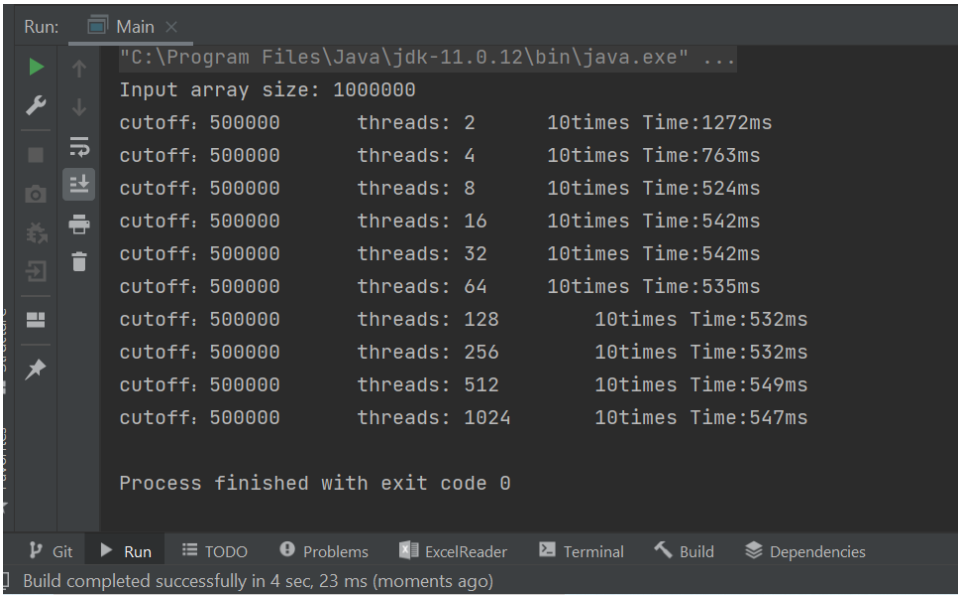
Approach 2 (based on number of threads)



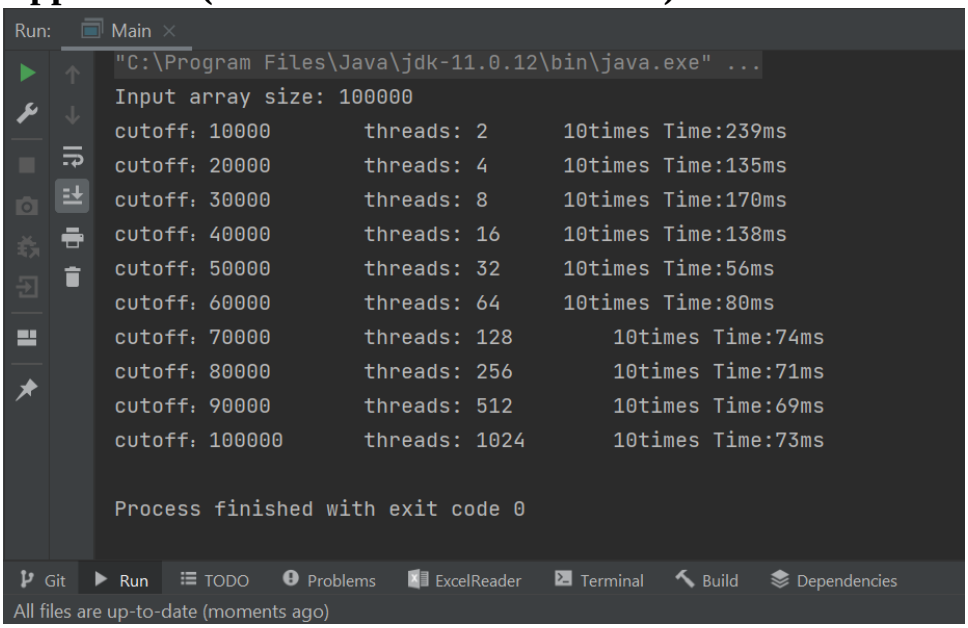

Run: Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 200000
cutoff: 100000 threads: 2 10times Time:344ms
cutoff: 100000 threads: 4 10times Time:145ms
cutoff: 100000 threads: 8 10times Time:114ms
cutoff: 100000 threads: 16 10times Time:119ms
cutoff: 100000 threads: 32 10times Time:109ms
cutoff: 100000 threads: 64 10times Time:115ms
cutoff: 100000 threads: 128 10times Time:107ms
cutoff: 100000 threads: 256 10times Time:101ms
cutoff: 100000 threads: 512 10times Time:102ms
cutoff: 100000 threads: 1024 10times Time:104ms
Process finished with exit code 0
Git Run TODO Problems ExcelReader Terminal Build Dependencies
All files are up-to-date (moments ago)

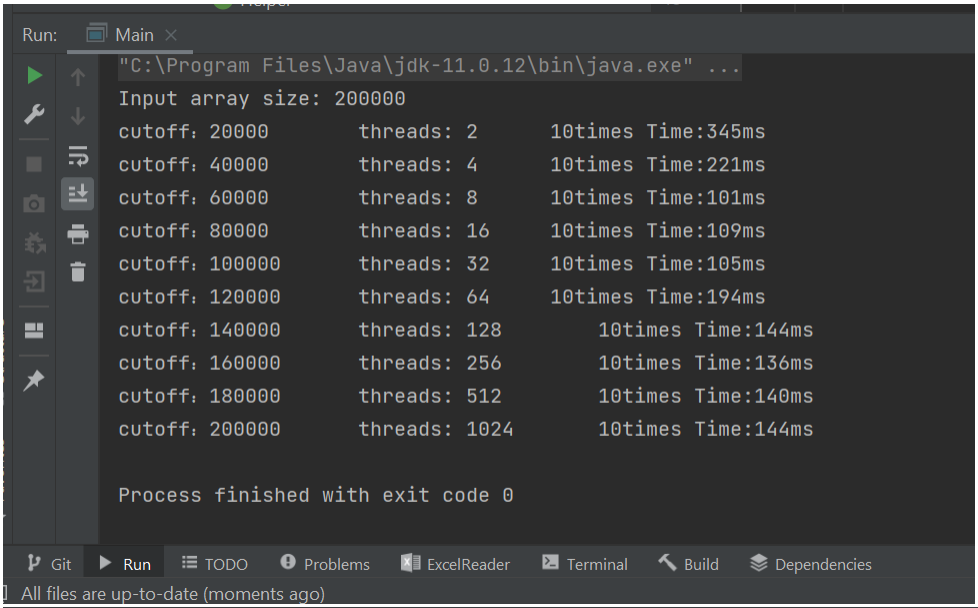

Run: Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 400000
cutoff: 200000 threads: 2 10times Time:605ms
cutoff: 200000 threads: 4 10times Time:303ms
cutoff: 200000 threads: 8 10times Time:220ms
cutoff: 200000 threads: 16 10times Time:209ms
cutoff: 200000 threads: 32 10times Time:202ms
cutoff: 200000 threads: 64 10times Time:196ms
cutoff: 200000 threads: 128 10times Time:201ms
cutoff: 200000 threads: 256 10times Time:201ms
cutoff: 200000 threads: 512 10times Time:205ms
cutoff: 200000 threads: 1024 10times Time:198ms
Process finished with exit code 0
Git Run TODO Problems ExcelReader Terminal Build Dependencies
All files are up-to-date (moments ago)

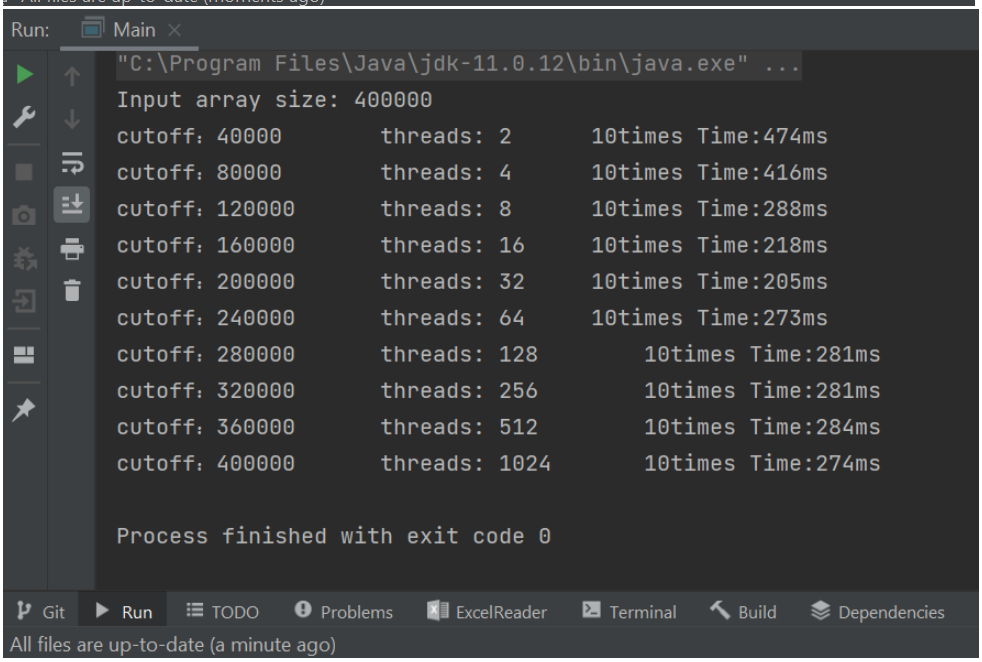

Run: Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 800000
cutoff: 400000 threads: 2 10times Time:1073ms
cutoff: 400000 threads: 4 10times Time:683ms
cutoff: 400000 threads: 8 10times Time:424ms
cutoff: 400000 threads: 16 10times Time:432ms
cutoff: 400000 threads: 32 10times Time:422ms
cutoff: 400000 threads: 64 10times Time:426ms
cutoff: 400000 threads: 128 10times Time:424ms
cutoff: 400000 threads: 256 10times Time:430ms
cutoff: 400000 threads: 512 10times Time:425ms
cutoff: 400000 threads: 1024 10times Time:431ms
Process finished with exit code 0
Git Run TODO Problems ExcelReader Terminal Build Dependencies
All files are up-to-date (moments ago)

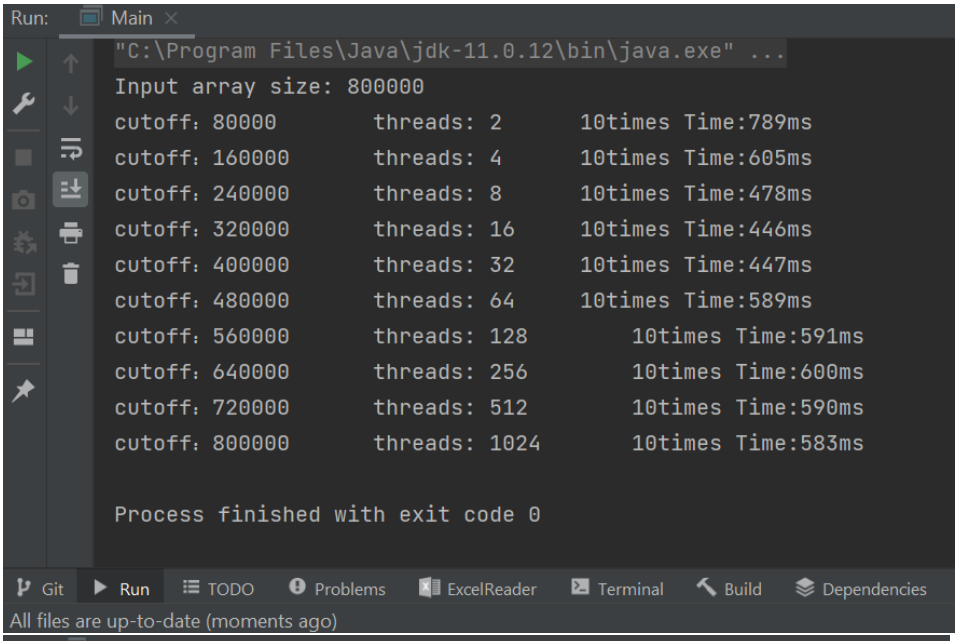


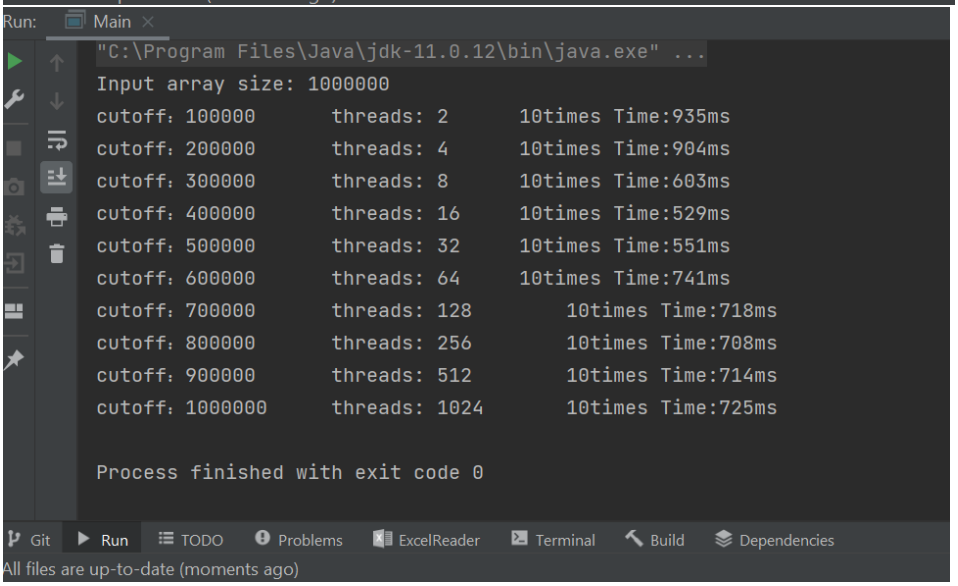
Approach 3 (based on cut off and threads)




Run:
Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 200000
cutoff: 20000 threads: 2 10times Time:345ms
cutoff: 40000 threads: 4 10times Time:221ms
cutoff: 60000 threads: 8 10times Time:101ms
cutoff: 80000 threads: 16 10times Time:109ms
cutoff: 100000 threads: 32 10times Time:105ms
cutoff: 120000 threads: 64 10times Time:194ms
cutoff: 140000 threads: 128 10times Time:144ms
cutoff: 160000 threads: 256 10times Time:136ms
cutoff: 180000 threads: 512 10times Time:140ms
cutoff: 200000 threads: 1024 10times Time:144ms
Process finished with exit code 0
Git
Run
TODO
Problems
ExcelReader
Terminal
Build
Dependencies
All files are up-to-date (moments ago)


Run:
Main
"C:\Program Files\Java\jdk-11.0.12\bin\java.exe" ...
Input array size: 400000
cutoff: 40000 threads: 2 10times Time:474ms
cutoff: 80000 threads: 4 10times Time:416ms
cutoff: 120000 threads: 8 10times Time:288ms
cutoff: 160000 threads: 16 10times Time:218ms
cutoff: 200000 threads: 32 10times Time:205ms
cutoff: 240000 threads: 64 10times Time:273ms
cutoff: 280000 threads: 128 10times Time:281ms
cutoff: 320000 threads: 256 10times Time:281ms
cutoff: 360000 threads: 512 10times Time:284ms
cutoff: 400000 threads: 1024 10times Time:274ms
Process finished with exit code 0
Git
Run
TODO
Problems
ExcelReader
Terminal
Build
Dependencies
All files are up-to-date (a minute ago)





- **Relationship Conclusion**

1. Parallel sort based merely on cut off
   Assuming the input array size is sufficiently large, with 5 different input array size, with the number of cut off increasing, according to the screenshots in Approach1, we can see that the consuming time can be least when the number of cut offs equals to the half of the array size. Also, there is a trend that when the cut off increase approaches to the half of the array size, the consuming time becomes less, when the cut off more than a half of the array size, the consuming time increase again.

2. Parallel sort based on threads

From the first approach, we conclude that when the number of cut offs equals to the half of the input array size, the consuming time is the least, and we take half of the array size as our cut off value and make it fixed in this approach. We can see from the screenshots in approach 2 that with the input array size increasing, the number of threads we need to get the least consuming time become less.

3. Parallel sort based on threads and cut off
   In this approach, we make number of threads and cut off as variables. From the screenshot in Approach 3, the least consuming time occurs approximately when the number of cut off equals to the half of the input array size and the thread number is 32.