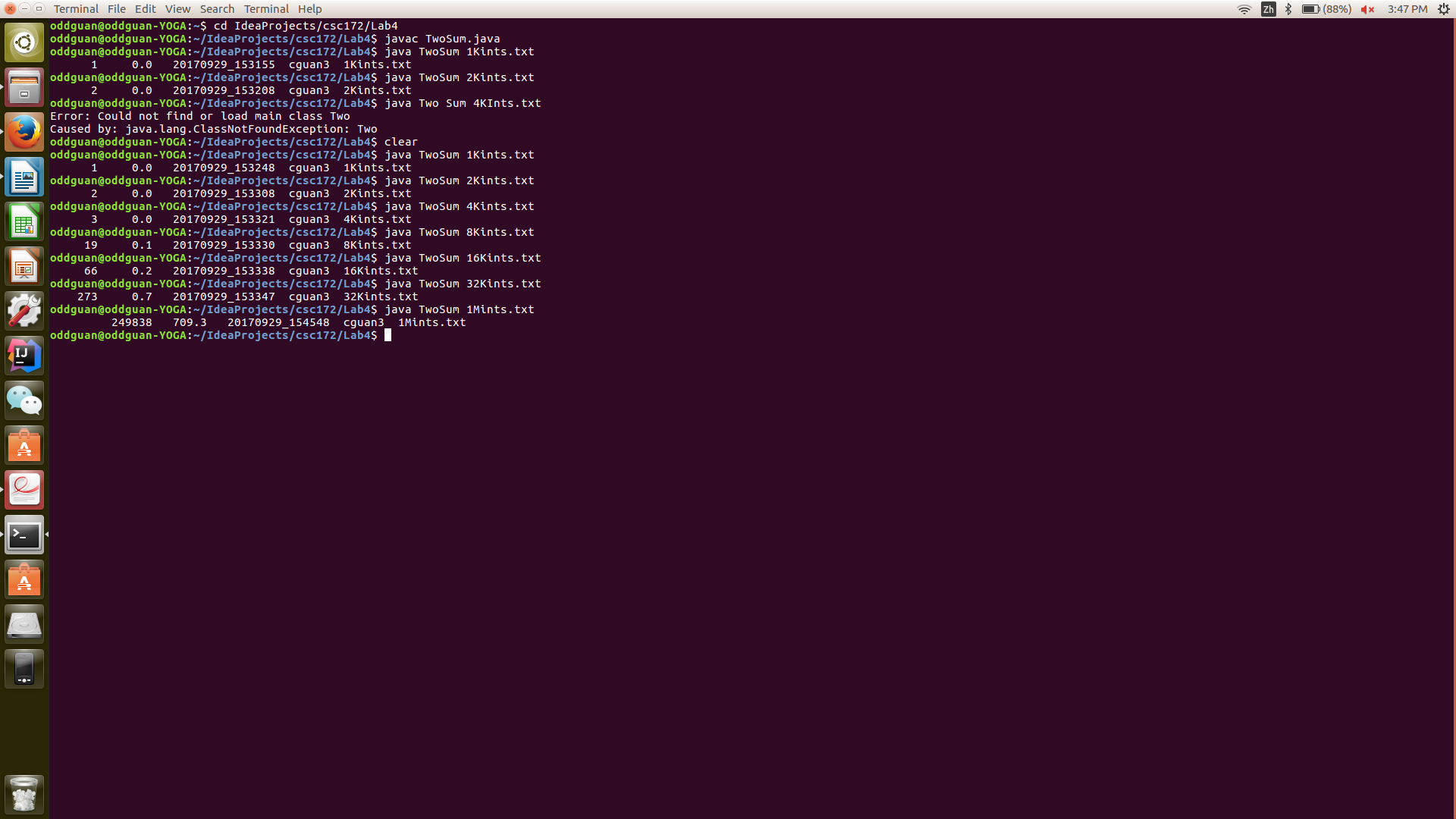
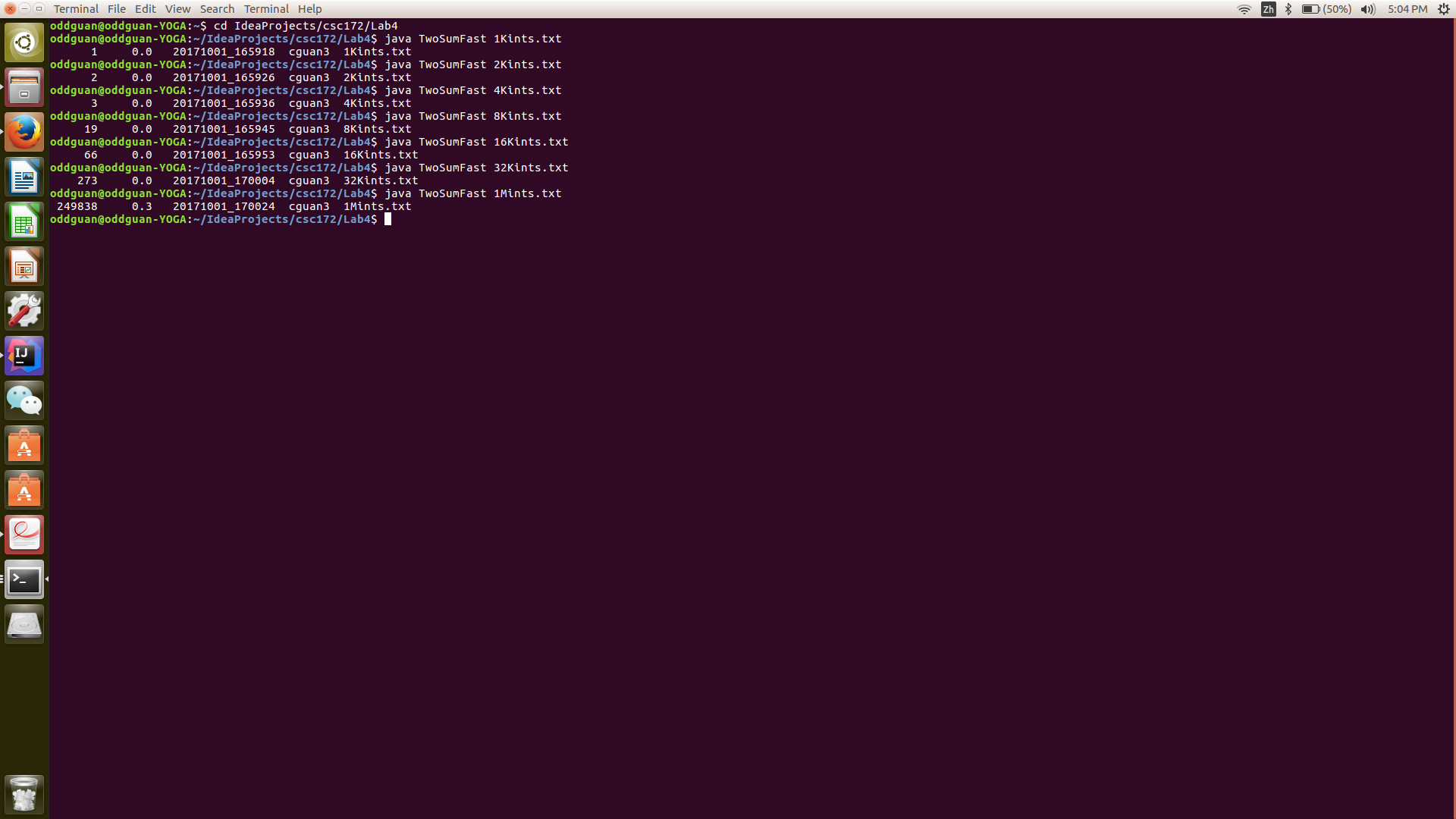
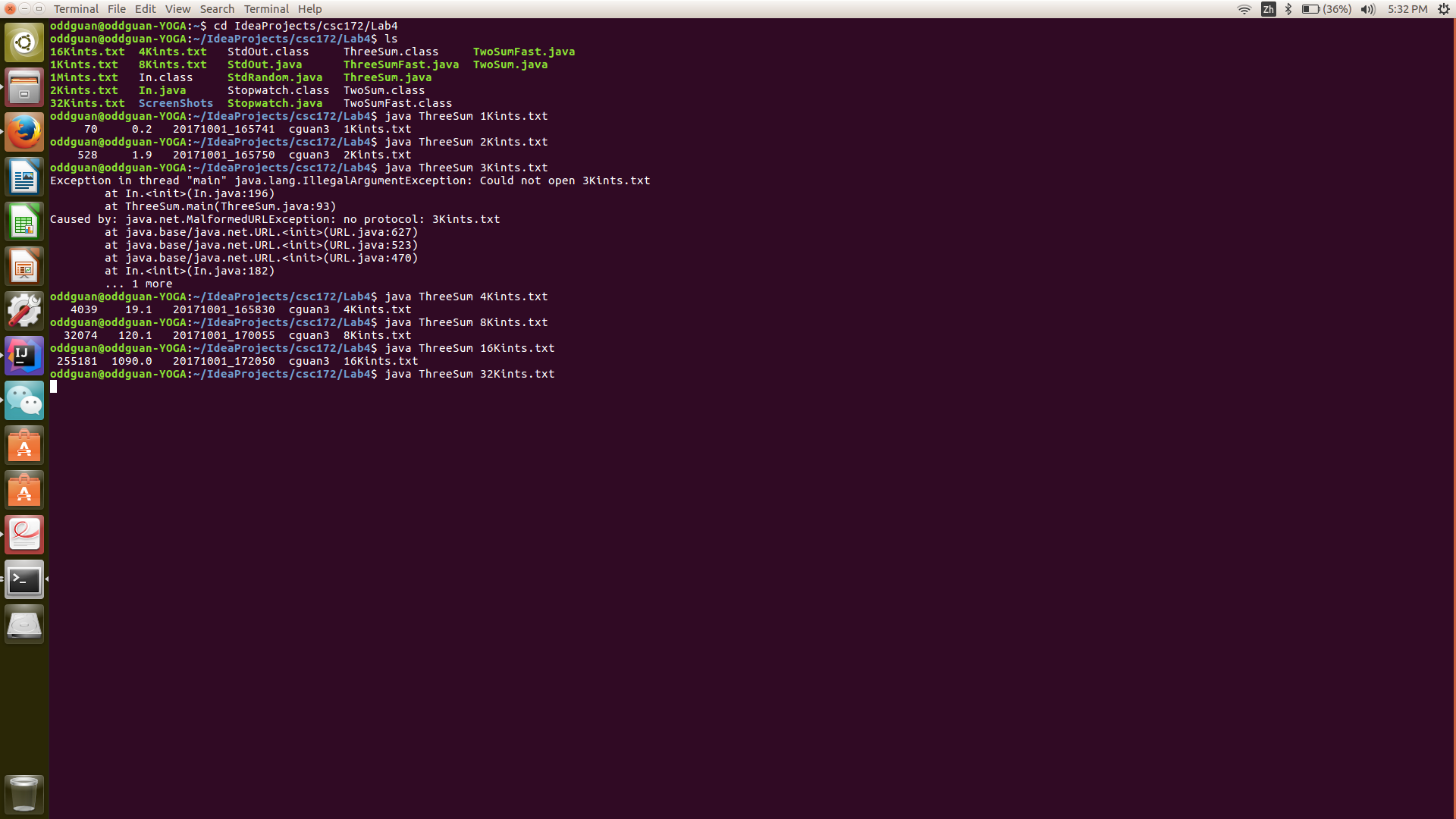
Task 1:

1. (TwoSum)

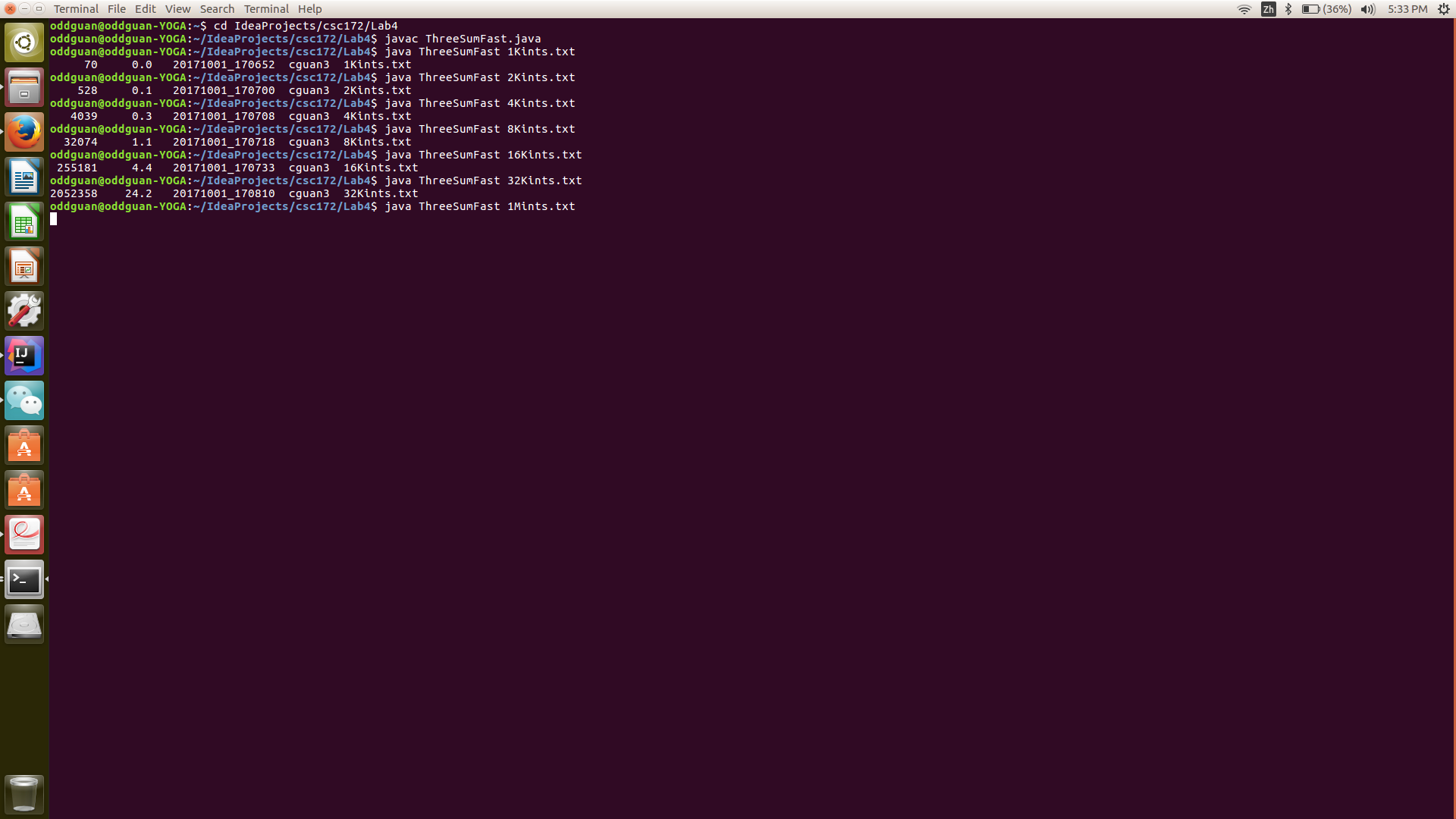
IT2.(TwoSumFast)



3.(ThreeSum)



4.(ThreeSumFast)



Task 2:

1.

2.

3.

4.

5.

For the TwoSum algorithm, when it comes to 1 million integers, the run time suddenly increases because it is a n^2 big O algorithm and it will increase very fast as n increases; whereas TwoSumFast is a nlog(n) algorithm and it takes a lot shorter to finish running.

6.

For the ThreeSum algorithm, it increases very fast when it comes to 8K and 16K, and I wasn’t even able to finish running 32K and 1M. Whereas ThreeSumFast takes much shorter time to finish (almost a flat linear line).

Task 3:

1. It takes 0.7 second to run the 32K file for the TwoSum method. We know the big O for this method is n^2, so the estimated run time for 1M file should be (1000000^2/32000^2)\*0.7 = 683.59. The actual run time, according to my screenshot, is 709.3 seconds, so it is pretty close.

2.Since this algorithm takes so much shorter time so that I don’t have the time it takes to run 1K, 2K and so on files, the only time we got in the terminal is the run time for 1Mints for this method and it is 0.3 second.

3. It takes 1090 seconds to run 16ints for the ThreeSum method, so the estimated time to finish running 32Kints, if the big O for this method is n^3, should be 1090\*(32000^3/16000^3)=8720 seconds. The run time for 1M ints should be 1090(1000000^3/16000^3)=266113281 seconds.

4. It takes 4.4 seconds to finish running 16Kints for ThreeSumFast method, so the estimated time to finish 32K, if the big O for this method is n^2\*log(n), should be 4.4\*(32000^2\*log(32000)/(16000^2)\*log(16000))=18.86 seconds. Our actual run time is 24 seconds, so it is pretty close. The estimated time for 1M ints should be 4.4\*(1000000^2\*log(1000000)/16000^2\*log(16000))=24529 seconds.