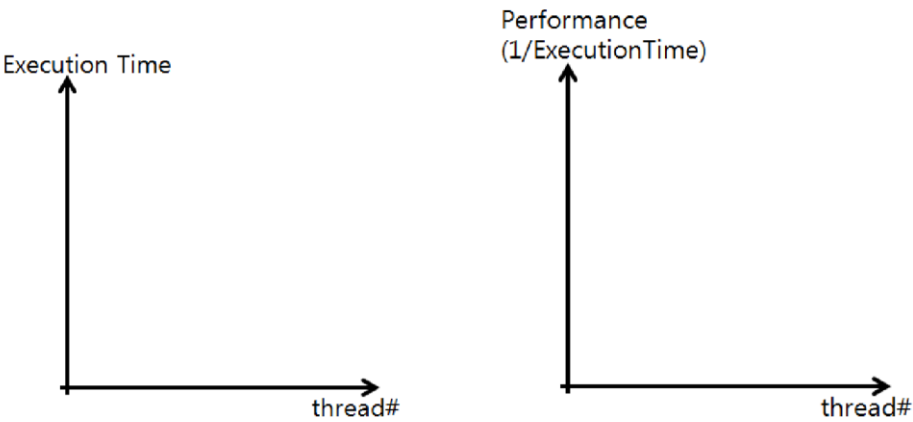


problem 1. In our lab. class, we looked at the JAVA program (ex4.java) that computes the number of ‘prime numbers’ between 1 and 200000. The java code (ex4.java) creates threads for parallel computation using static load balancing approach. However, The naive implementation of ex4.java may not give satisfactory performance because of bad load balancing. The problem is that (i) higher ranges have fewer primes and (ii) larger numbers take longer time to test. Therefore thread workloads may become uneven and hard to predict. For better performance, we consider dynamic load balancing approach where each thread takes a number one by one and test whether the number is a prime number.

- (i) Modify ex4.java to adopt dynamic load balancing instead of static load balancing and submit the modified JAVA code (name it "ex4_dynamic.java"). Your dynamic load balancing code should use JAVA synchronization method. Your code also should print the (1) execution time of each thread and (2) execution time for the entire thread computation. When writing your JAVA code, NUM_THREAD variable should be used as a constant value in the program (just like ex4.java).
- (ii) Write a document that reports the parallel performance of your code. The graph that shows the execution time when using 1, 2, 4, 6, 8, 10, 12, 14, 16 threads. There should be at least two graphs, one for static load balancing and the other for dynamic load balancing. Your document also should mention which CPU (dualcore? or quadcore?, clock speed) was used for executing your code.



exec time	1	2	4	...	16
static					
dynamic					

performace (1/exec time)	1	2	4	...	16
static					
dynamic					