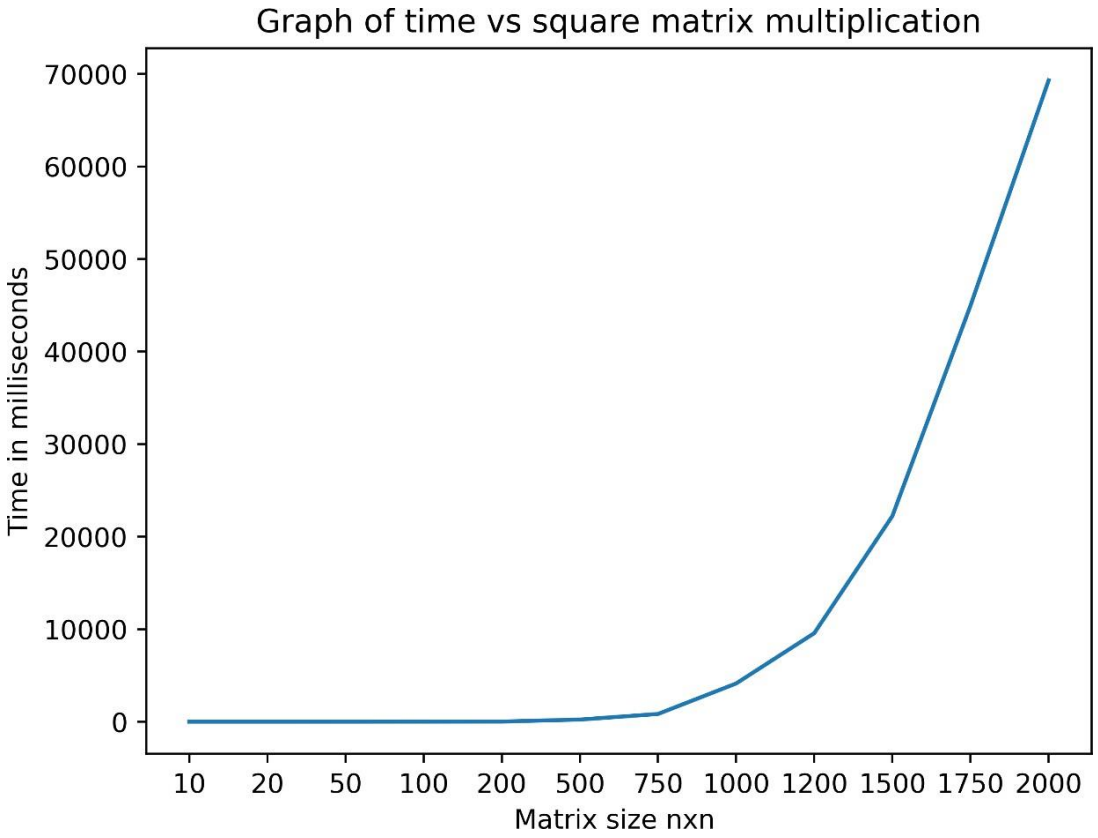
EEM 480 Homework-1 Report

The purpose of this homework was to demonstrate the effects of matrix size in multiplication on time and find the prime numbers for the given matrices and the multiplied matrices result. The homework was developed in Apache NetBeans IDE 12.0 using JDK8-301.

At beginning of the code Cmatrix.java, 2-dimensional long list object respectively called elements [][] and temp [][] declared. The list called elements holds the whole matrix with rows and columns and the temp list holds the values for the result matrix after multiplication. Int variables called row and col holds the values of the constructed matrix's rows and columns. Random object created from java.util.Random library for appending random values from 0 to 10000 as given in the homework instruction. Two class constructor method was used. If the user wants to create an object without giving parameters of row and col sizes then the second constructor block (Starting from Line[24]) will be used with the pre-initialized values for row and col. This is called polymorphism. AssignRandom method created with a nested loop to iterate in rows and columns in the matrix. In the inner loop, random values are assigned to the elements list with the given indices in the loop. The method called printMatrix was created for printing elements variable of the given object. The method uses a nested loop for iterating rows and columns and in the inner loop print statement with the blank space between numbers printed out. PrintMatrixWithPrime method's algorithm used for printing the prime numbers in matrix with " * ". The long half variable is used for dividing the number by two. The main reason for that is to check every number in the matrix to half of the number. Because there is no need for checking the number from 2 to number itself for detecting prime numbers. This algorithm saves time. After that the if the number can be divided by any number between 2 and half of it then the algorithm set the prime flag to 1. If prime = 1 then the system prints the number without "*" prime = 1 means the number is not prime. Else statement says that if the flag is not 1 then it is a prime number and prints "* "with the number. The last method called multiply matrices takes another CMatrix object to multiply with the matrix used in the method. A new object created from System.currentTimeMillis called startTime to measure the time of multiplication. Temp list created with the m4.row and the Multiplicand.col. The reason for that is the result of multiplication changes the matrix size. Three nested loops were used for demonstrating matrix multiplication. This.elements[0][0] and the Multiplicand.elements [0][0] multiplied and appended in temp [0][0] after iterations we can see that with this algorithm this.elements rows multiplied to the Multiplicand.columns and the total result appended on the temp list's index. After that, multiplication consumed time calculated and the result printed accordingly to the given instruction in the homework document. At last, the function returns an object and overwrites to the already generated object. Multiplications tested with (10x10 20x20 50x50 100x100 200x200 500x500 750x750 1000x1000 1200x1200 1500x1500 1750x1750 2000x2000) matrix sizes and the result is transformed in to the Graph 1.

I observed that the time consumption grows exponentially by increasing the size of the matrix.



Graph 1 *Graph of consumed time with the increasing in square matrix size*