In space provided below attach the report (PDF) that contains the following section (section headings in bold text).

1. Availability. A URL for the source code where your code is available. Please ensure that the code is documented. (0.5 point)

https://github.com/kallepaa/programming-assigment-01

2. Programming Language. Details on the choice of programming language you used, and the libraries. (0.5 point)

I choose C# because I’m familiar with it. I used .Net Core 6.0 version. I did not use any external libraries. Program can be compiled and run using command “dotnet run –release”. Needs dotnet installed in machine.   https://dotnet.microsoft.com/en-us/download

3. Methodology. Details of how the matrix are created and multiplied, and the techniques for evaluating the performance. (2 points).

I used associativity to first multiply B and C and then that result with A (A(BC)) to reduce amount operations.

When matrixes are

A = 10^6 x 10^3

B = 10^3 x 10^6

C = 10^6 x 1

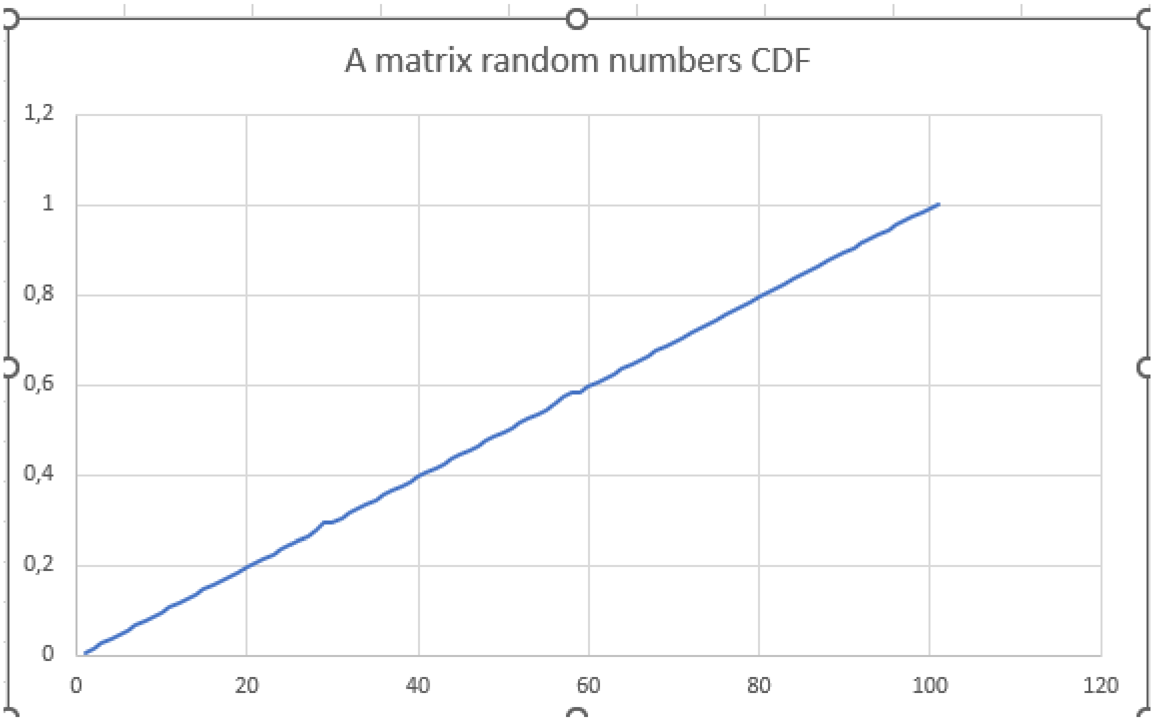
Then

ABC has 1 001 000 000 000 000 operations

A(BC) has 2 000 000 000 operations

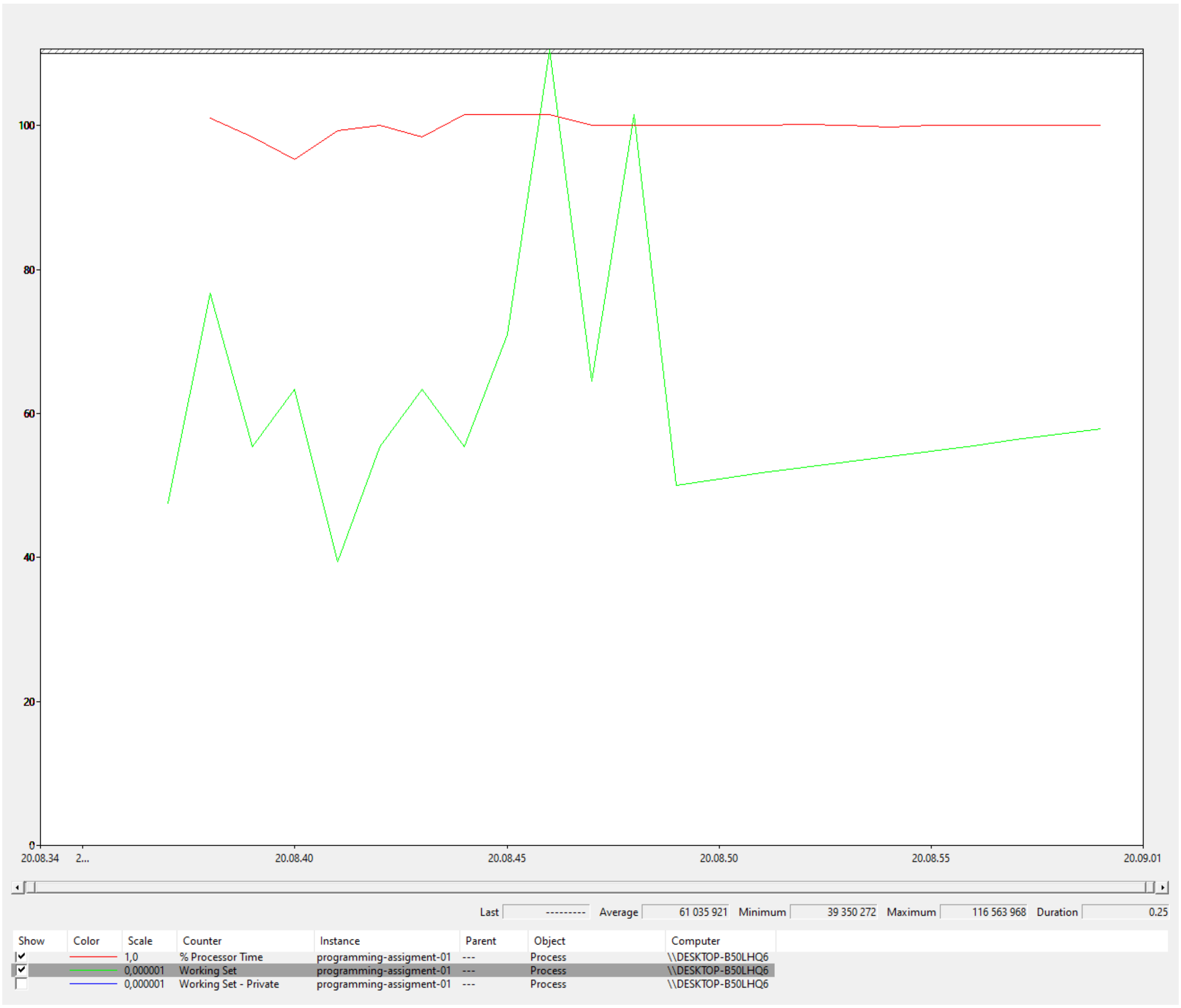
ABC has 50 0500 times more operations than ABC

4. Dataset. Plot a (empirical) cumulative distribution function (CDF) of all the values present in matrix A. What can you infer from the CDF? (2 points)



I can see that random number generator generates number from uniform distribution in range 0 to 1.

5. Evaluation. Plot the time evolution of the memory usage, and CPU (and/or GPU) usage of your solution. What do you infer from the plots? (2 points)



Last --------- Average 61 035 921 Minimum 39 350 272 Maximum 116 563 968 Duration 0.25

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟥	1,0	% Processor Time	programming-assigment-01	---	Process	\\DESKTOP-B50LHQ6
✓	🟩	0,000001	Working Set	programming-assigment-01	---	Process	\\DESKTOP-B50LHQ6
	🟦	0,000001	Working Set - Private	programming-assigment-01	---	Process	\\DESKTOP-B50LHQ6

I can see from plot that processor utilization is ~ 100% and memory usage is jumping up and down.

6. Discussion. Detail the various challenges encountered and how you addressed those challenges. If you are unable to obtain D, what is the largest size of the matrices for which you were able to compute D. (3 points)

First, I tried to directly execute ABC and get out of memory. Then I decided to load bigger matrixes A and B one row at the time. But then remember associative and changed order of multiplication to A(BC). This reduced operations and size of required matrix for intermediated result. I did not have time to test if program works after change of order even not to do row by row loading for A and B matrixes.

Probably memory utilization can be enchantment by using more advanced techniques in .Net, but I id does not have time to evaluate them. What I would be interest is optimize how GC behaves when large amount memory is allocated for the application process.