

# CMSC 180 Laboratory Research Problem 1

## Computing the Serial Solution to the Min-Max Transformation of Elements in a Column of a Matrix

Aldrine Drebb L. Cristobal  
CD-3L

January 2026

### 1 Introduction

Min-Max Transformation (MMT) is a type of matrix arithmetic wherein the elements are subjected to the following equation.

$$T_{ij} = (X_{ij} - \min(X_j)) / (\max(X_j) - \min(X_j))$$

where min and max are used to identify the minimum and maximum element of the jth column of X.

### 2 Methodology

The program is ran using a VirtualBox machine with an Ubuntu v24.04.3 operating system. The VirtualBox is allocated with 8192 MB RAM and 2 Processors.

The code is written using C language and developed using Visual Studio Code. The researcher made use of timespec data structure to capture the time with high accuracy up to nanoseconds.

### 3 Research Questions

#### 3.1 The complexity of solving the Min-Max Transformation of an N \* N square matrix X

The time complexity of Min-Max Transformation is  $O(n^2)$ . Two for loops are used to access and rewrite each element, one for traversing through its rows and one for traversing through its columns.

#### 3.2 The feasibility of solving the Min-Max Transformation with N greater than 20,000

Even though the time it took to solve matrices with N less than and equal to 20,000 is so low, the memory it takes to process them grows larger than the memory could handle. An attempt at solving N = 50,000 froze the VirtualBox prompting an emergency reset.

Better management of memory may be able to bypass this limitation such as making use of files to temporarily store data or making use of greater amount of memory.

#### 3.3 The average run time and time complexities using the minimum and maximum N

Seen in the first figure is the graph showcasing the average run time and time complexities using the minimum (100) and maximum (20,000) N. The following formula is used:

$$Theoretical_i = (N_i / N_{max|min})^2 * Average_{N(max|min)}$$

The average run time closely follows the theoretical run time using the maximum number of N while using the minimum slightly deviates around N = 8,000.

#### 3.4 Adjustments and Improvements

The orientation of the data, column-major, provide a lot more to be improved upon. Pre-processing it as its

transpose may provide faster accessing of minimum and maximum data points. The finished row can also be removed from the matrix and transferred to another storage such as a text file to ease the memory of its load.

After making use of the transposed version of the matrix, it can be further optimized by making use of division less often as it has a large overhead.  $1/(\min(X_i) - \max(X_i))$  can be precomputed and multiplied to each  $X_{ij} - \min(X_i)$ .

## 4 Appendix

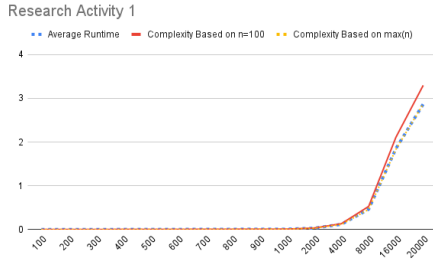


Figure 1: A graph comparing Average Run Time and Theoretical Time Complexity using the lowest and highest N that ran successfully.

N	Run 1	Run 2	Run 3	Average Runtime	Complexity Based on n=100	Complexity Based on max(n)
100	0.00002549	0.00002569	0.00002608	0.00002595	0.00002595	0.000716870304
200	0.000382763	0.00034739	0.000303303	0.00034385	0.00034385	0.0002087481952
300	0.000716098	0.000702044	0.000708799	0.000708799	0.000716098	0.0004481832614
400	0.001199473	0.001454878	0.0017384051	0.001344111	0.001344111	0.001148992625
500	0.001919430	0.001625466	0.001880096	0.001761721	0.001880096	0.001762173076
600	0.002676136	0.002976296	0.003289348	0.002976793	0.002976793	0.002588773406
700	0.004089832	0.004376827	0.004718826	0.004376826	0.004376826	0.003520648913
800	0.0046975168	0.005068589	0.005748899	0.005068589	0.005068589	0.00468779489
900	0.006379798	0.006897357	0.008465272	0.006897357	0.006897357	0.0065888659152
1000	0.00762477	0.00777671	0.007584650	0.007584650	0.00777671	0.007168702964
2000	0.02952550	0.03209157	0.029353112	0.029604040	0.03209157	0.02867481952
4000	0.117624483	0.176274853	0.127432632	0.118447793	0.176274853	0.1148826262
8000	0.462624860	0.467913999	0.467616815	0.462624860	0.467913999	0.4687979499
16000	1.88898988	1.83655888	1.847284874	1.858971490	1.88898988	1.83655888
20000	2.89628125	2.896280807	2.877057752	2.896280807	2.89628125	2.896280807

Figure 2: The data set gathered during the experimentation.