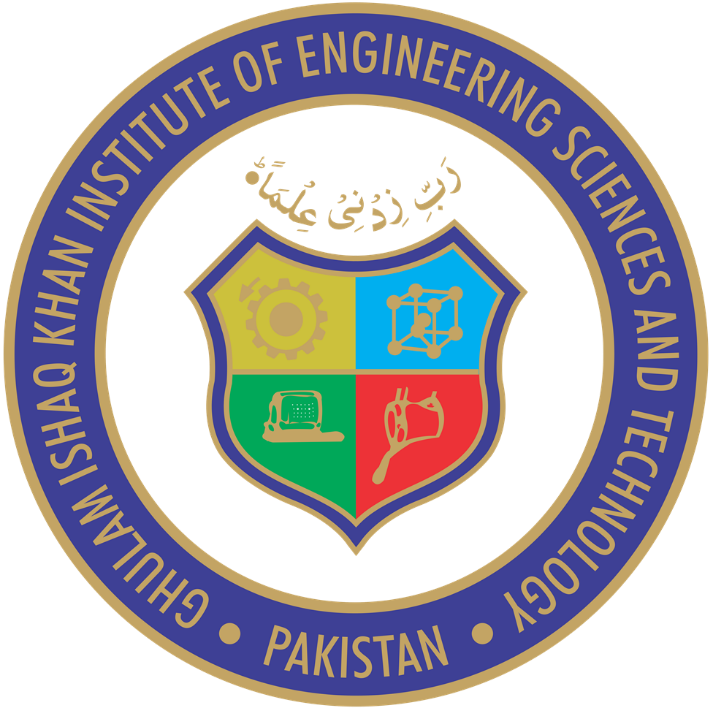
**CS221 Data Structures and Algorithms**

**Project Report**

Ghulam Ishaq Khan Institute of Engineering Sciences & Technology, Topi, Swabi 23460, Pakistan

****

**Group Members:**

* **2021345** (Muhammad Ali Khan)
* **2021638** (Syed Muhammad Haider Zaidi)
* **2021083** (Muhammad Talha Imtiaz)

**Table of Contents:**

**1. Project Description---------------------------------Page.3**

**2. Comparison of Task 2 & 3------------------------Page.8**

**3. Task Distribution-----------------------------------Page.9**

**4. References--------------------------------------------Page.9**

**Project Description:**

**Task 1:**

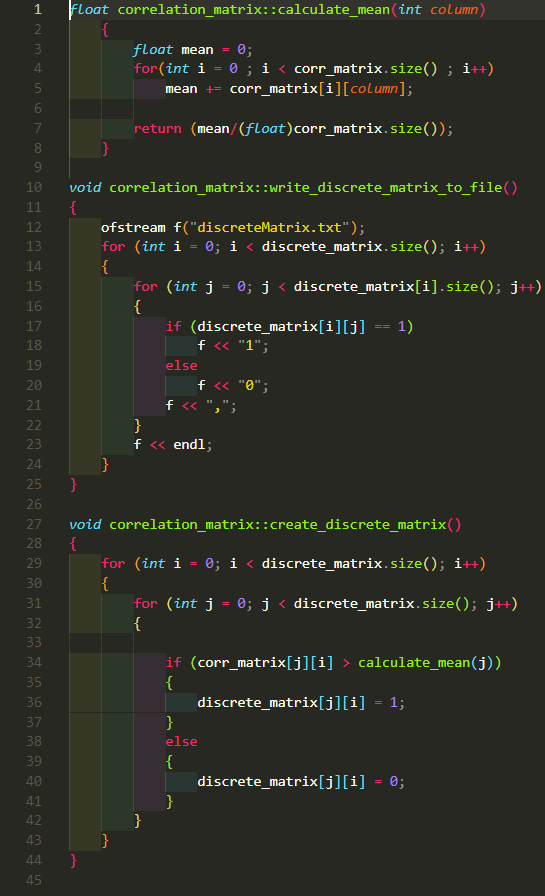
We created a correlation (or similarity) matrix using Pearson’s coefficient on each row of the dataset. The size of the matrix is the nxn (where n is the number of rows in the dataset).

**Text

Description automatically generated**

**Fig. 1.** Code for calculating the Pearson’s coefficient and correlation matrix

The correlation matrix is then discretized by making all values in each of its column lower than the mean of that respective column 0, and 1 if higher.

**** **Chart

Description automatically generated with medium confidence**

**Fig. 2 and 3.** Code and Bitmap for the discretized matrix

To display the color-coded image of the similarity matrix, we find the maximum value in each of its columns. Then each value in the column is divided by the max value, after which it is multiplied with 255. The result would be the RGB value of each pixel in the bitmap.

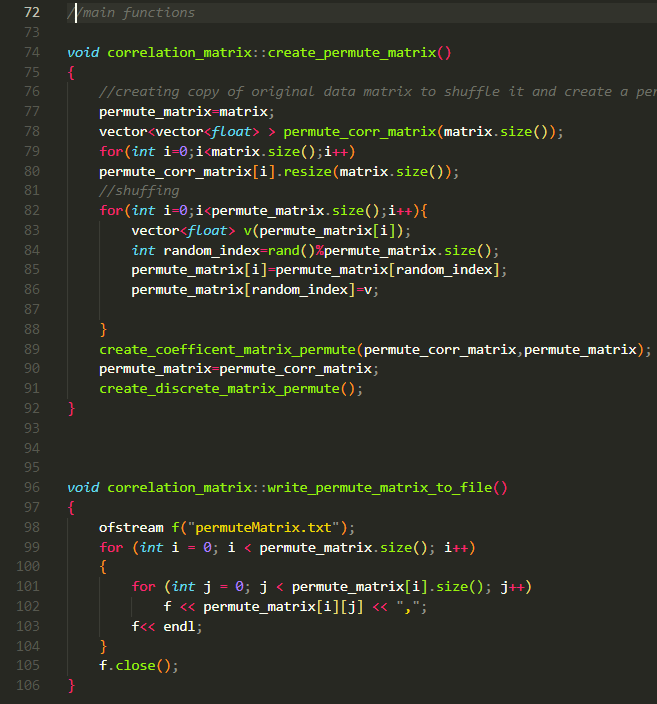
 **Background pattern

Description automatically generated**

**Fig. 4 and 5.** Code and Bitmap for the similarity matrix

**Task 2:**

The Data Matrix is permuted by shuffling individual rows in the dataset. After which we calculate its correlation matrix which contains RGB values for the color-coded bitmap image.

**** A picture containing electronics, circuit

Description automatically generated

**Fig. 6 and 7.** Code and Bitmap for the permuted matrix

Then we calculate the signatures of each row of the permuted data matrix and then store it as a column in it, calling it signature matrix. This vector along with the permuted matrix is called the similarity matrix. The matrix is then sorted based on signature and the task 1 functions were applied to the rearranged signature matrix.

**** **A picture containing outdoor

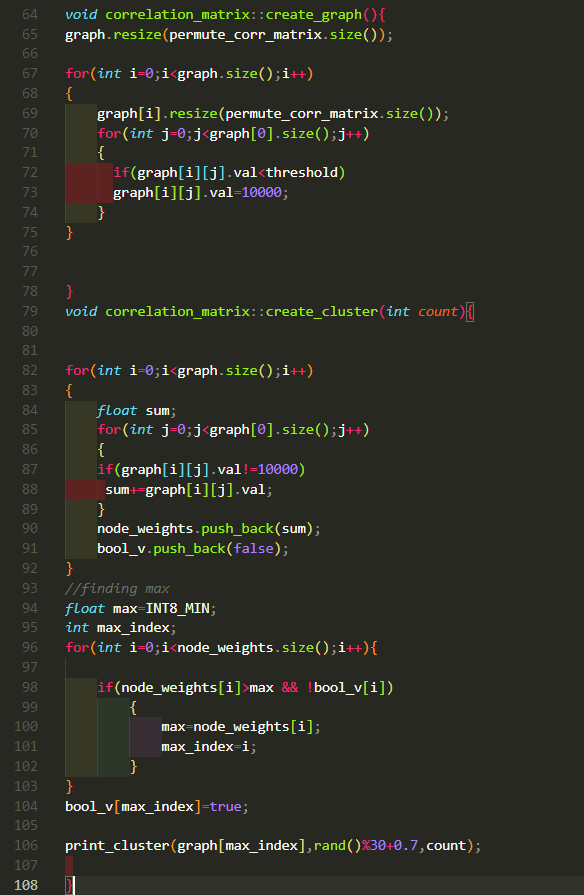
Description automatically generated**

**Fig. 8 and 9.** Code and Bitmap for the rearranged signature matrix

**Task 3:**

The permuted dataset is then converted into weighted graph by calculating is correlation matrix. Then all edges with weights below a certain threshold are removed by making them a 0.

The nodes and then are assigned weights by summing all the weights of the edges connected to each node. The node with the highest weight is then found and all edges connected to it shown as a cluster. It is then displayed as a bitmap and them removed from the graph. This step is repeated until no cluster is left.

 **Background pattern

Description automatically generated**

**Fig. 10 and 11.** Code and Bitmap for creating a cluster

**Comparison of Task 2 & 3:**

In Task 2 we recover the image clusters using the signature technique.

In Task 3, however, we visualize those recovered image clusters by weighted graph technique i.e., looping through clusters of every node in descending order.

**Task Distribution:**

Talha Imtiaz: Devised logic for Task 3 and made the bitmaps

Ali Khan: Devised logic for Task 2 and coded all tasks

Haider Zaidi: Devised logic for Task 1 and made the report

**References:**

1. <https://youtu.be/vqT5j38bWGg> (Creating a Bitmap Image (.bmp) using C++)