```java
// Fundamentals of Genomics & Proteomics
// Lab 02
// Shamim Bin Zahid
// Roll: 43

import java.util.List;
import java.util.ArrayList;

public class Main{
    // Global Variables Here Instead of Input
    public static double[][] distanceMatrix = {
        {  0.0, 22.0, 39.0, 39.0, 41.0 },
        { 22.0,  0.0, 41.0, 41.0, 43.0 },
        { 39.0, 41.0,  0.0, 18.0, 20.0 },
        { 39.0, 41.0, 18.0,  0.0, 10.0 },
        { 41.0, 43.0, 20.0, 10.0, 0.0  }
    };
    public static int matrixSize = distanceMatrix.length;
    public static int[][] minimunValues = new int[matrixSize][2];
    public static List < Integer > clusterSize = new ArrayList < Integer > ();

    // Updating the matrix reduced in every step
    public static double[][] updateMatrix(double[][] distanceMatrix, int i) {
        double[][] updatedMatrix = new double[i][i];
        int[] position = new int[2];
        double minValue = 100;
        for (int j = 0; j < distanceMatrix.length; ++j) {
            for (int k = j + 1; k < distanceMatrix.length; ++k) {
                if (minValue > distanceMatrix[j][k]) {
                    minValue = distanceMatrix[j][k];
                    position[0] = j;
                    position[1] = k;
                }
            }
        }
        minimunValues[i][0] = position[0];
        minimunValues[i][1] = position[1];
        int first = 0, second = 1;
        int count = matrixSize - i -1;
        for (int j = 0; j + 1 < distanceMatrix.length; ++j) {
            for (int k = j + 1; k + 1 < distanceMatrix.length; ++k) {
                if (k == updatedMatrix.length - 1) {
                    double temp = (distanceMatrix[j][k] * 1.0 + distanceMatrix[j][k + 1] * (count + 1)) / (2 + count);
                    updatedMatrix[j][k] = updatedMatrix[k][j] = temp;
                } else {
                    updatedMatrix[j][k] = updatedMatrix[k][j] = distanceMatrix[j][k];
                }
                second++;
            }
            updatedMatrix[j][j] = 0;
            first++;
        }
        clusterSize.set(position[1], clusterSize.get(position[1]) + clusterSize.get(position[0]));
        clusterSize.remove(position[0]);
        return updatedMatrix;
    }
    // Driver code
    public static void main(String args[]) {
        for(int i = 0; i < matrixSize; i++) clusterSize.add(1);
        int stepCount = 0;
        for (int i = matrixSize-1; i > 0; --i) {
            System.out.println("Step "+ stepCount++);
            for (int j = 0; j < distanceMatrix.length; ++j) {
                for (int k = 0; k < distanceMatrix.length; ++k) {
                    System.out.format("%.2f ", distanceMatrix[j][k]);
                }
                System.out.println();
            }
            System.out.println();
            if(i!=1) distanceMatrix = updateMatrix(distanceMatrix, i);
        }
    }
}

```