**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе**

по предмету «СИАОД»

на тему:

«Методы сортировки»

Выполнил: студент группы

Митрохин Ярослав Игоревич

Руководитель:

Кутейников Иван Александрович

Москва 2020

*Цель работы:* реализовать заданный метод сортировки числовой матрицы в соответствии с индивидуальным заданием.

*Ход работы:*

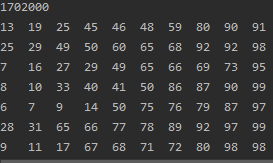
1. *Код программы*

import java.util.Random;  
import java.util.Arrays;  
  
public class MAIN {  
 static int *rows* = 10;//строчки  
 static int *cols* = 10;//столбцы  
 public static void main (String[] args)  
 {  
 int[][] array = new int[*rows*][*cols*];  
 for (int i = 0; i < *rows*; i++){  
 for (int j = 0; j < *cols*;j++) {  
 array[i][j] = (int) (Math.*random*()\*100);  
 }  
 }  
 long time = System.*nanoTime*();  
 *shSort*(array);  
 System.*out*.println((System.*nanoTime*()-time));  
 time = 0 ;  
 long time1 = System.*nanoTime*();  
 *standartSort*(array);  
 System.*out*.println((System.*nanoTime*()-time1));  
 time1 = 0 ;  
 long time2 = System.*nanoTime*();  
 *quickSort*(array, 0,9);  
 System.*out*.println((System.*nanoTime*()-time2));  
 time2 = 0 ;  
 //printMatr(array);  
 }  
 public static void standartSort(int[][] array){  
 for (int i = 0; i < *rows*;i++){  
 Arrays.*sort*(array[i]);  
 }  
 }  
 public static void printMatr(int[][] array){  
 for(int i=0;i<*rows*;i++){  
 for(int j=0;j<*cols*;j++) {  
 if (j==*cols*-1)  
 System.*out*.println(array[i][j]);  
 else  
 System.*out*.print(array[i][j]+"\t");  
 }  
 }  
 }  
 public static void quickSort(int[][] array, int low, int high)  
 {  
 if (array.length == 0)  
 return;  
 if (low >= high)  
 return;  
 int middle = low + (high - low)/2;  
 for (int i = 0; i < *rows*; i++) {  
 int op = array[i][middle];  
 int k = low, l = high;  
 while (k<l)  
 {  
 while (array[i][k] < op)  
 {  
 k++;  
 }  
 while (array[i][l] > op)  
 {  
 l--;  
 }  
 if (k <= l)  
 {  
 int temp = array[i][k];  
 array[i][k] = array[i][l];  
 array[i][l] = temp;  
 k++;  
 l--;  
 }  
 }  
 if (low < l)  
 *quickSort*(array,low,l);  
 if (high > k)  
 *quickSort*(array, k, high);  
 }  
 }  
  
 public static void shSort(int[][] array ) {  
 int h = 1;  
 while (h\*3 < array.length)  
 h=h\*3+1;  
 while (h>=1){  
 *shellSort*(array, h);  
 h = h/3;  
 }  
 }  
 public static void shellSort(int[][] array, int h) {  
 int length = *cols*;  
 for (int k = 0; k < *rows*; k++) {  
 for (int i = h; i < length; i++) {  
 for (int j = i; j >= h; j = j - h) {  
 if (array[k][j] < array[k][j - h]) {  
 int temp = array[k][j];  
 array[k][j] = array[k][j - h];  
 array[k][j - h] = temp;  
 }  
 else  
 break;  
 }  
 }  
 }  
 }  
 public static void swap(int[][] array, int i, int j)  
 {  
 for(int k = 0;k < *rows*; k++) {  
 int temp = array[k][i];  
 array[k][i] = array[k][j];  
 array[k][j] = temp;  
 }  
 }  
  
}

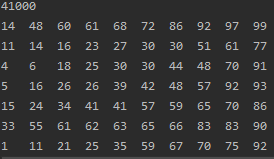
1. *Результаты программы*
2. *Отсортированная матрица методом стандартной сортировки:*

**

1. *Отсортированная матрица методом быстрой сортировки:*

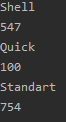
**

1. *Отсортированная матрица методом Шелла:*

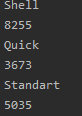
**

1. *Сравнение времени выполнения всех методов*

Матрица 10 на 10:



Матрица 100 на 100:

**

*Выводы:* реализовал заданные методы сортировки числовой матрицы в соответствии с индивидуальным заданием. Сравнил время выполнения всех методов сортировки. По результатам видно, что быстрая сортировка быстрее при любой размерности матрицы. Сортировка методом Шелла намного быстрее при матрицы размерности меньше 50, далее сортировка Шелла становится медленнее всех.