**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра «Информатики»

**Отчет по лабораторной работе №1**

Выполнил: студент группы БВТ1901

Кускова А. Е.

Руководитель:

Мелехин А.

Москва 2021

**Задание:**

Реализовать методы сортировки строк числовой матрицы в соответствии с заданием. Оценить время работы каждого алгоритма сортировки и сравнить его со временем стандартной функции сортировки. Испытания проводить на сгенерированных матрицах.

**Решение:**

Реализуем заданные методы сортировки, используя язык программирования Java. Встроим в программу рандомайзер матрицы размером 50\*50.

Код для пирамидальной сортировки:

import java.util.Random;

public class HeapSort {

public static void main(String[] args) {

int[][] array2d = new int[50][50];

Random rand = new Random();

for (int i = 0; i < 50; i++)

for (int j = 0; j < 50; j++)

array2d[i][j] = rand.nextInt(1012);

for(int[] m : toMatrix(sort(toArray(array2d)))) {

for(int i : m){

System.out.print(i + "\t");

}

System.out.println();

}

System.out.println();

long time = System.currentTimeMillis();

sort(toArray(array2d));

System.out.println(System.currentTimeMillis() - time);

}

public static int[] toArray(int[][] arr){

int[] flat = new int[50 \* 50];

int ctr = 0;

for (int row = 0; row < 50; row++) {

for (int col = 0; col < 50; col++) {

flat[ctr++] = arr[row][col];

}

}

return flat;

}

public static int[][] toMatrix(int[] arr){

int [][] numbers = new int [50][50];

int m = 0;

for(int i = 0 ; i < 50 ; i++)

for(int j = 0 ; j < 50; j++)

numbers[i][j] = arr[m++];

return numbers;

}

// Процедура для преобразования в двоичную кучу поддерева с корневым

// узлом i, что является индексом в arr[]. n - размер кучи

public static void heapify(int arr[], int n, int i)

{

int largest = i; // Инициализируем наибольший элемент как корень

int l = 2\*i + 1; // левый потомок = 2\*i + 1

int r = 2\*i + 2; // правый потомок = 2\*i + 2

// Если левый дочерний элемент больше корня

if (l < n && arr[l] > arr[largest])

largest = l;

// Если правый дочерний элемент больше, чем самый большой элемент на

// данный момент

if (r < n && arr[r] > arr[largest])

largest = r;

// Если самый большой элемент не корень

if (largest != i)

{

int swap = arr[i];

arr[i] = arr[largest];

arr[largest] = swap;

// Рекурсивно преобразуем в двоичную кучу затронутое поддерево

heapify(arr, n, largest);

}

}

public static int[] sort(int arr[])

{

int n = arr.length;

// Построение кучи (перегруппируем массив)

for (int i = n / 2 - 1; i >= 0; i--)

heapify(arr, n, i);

// Один за другим извлекаем элементы из кучи

for (int i=n-1; i>=0; i--)

{

// Перемещаем текущий корень в конец

int temp = arr[0];

arr[0] = arr[i];

arr[i] = temp;

// Вызываем процедуру heapify на уменьшенной куче

heapify(arr, i, 0);

}

return arr;

}

}

Код для быстрой сортировки:

import java.util.Random;

public class QuickSort {

public static void main(String[] args) {

int[][] array2d = new int[50][50];

Random rand = new Random();

for (int i = 0; i < 50; i++)

for (int j = 0; j < 50; j++)

array2d[i][j] = rand.nextInt(1012);

/\*for (int i = 0; i < 50; i++) {

for (int j = 0; j < 50; j++) {

System.out.print(array2d[i][j] + "\t");

}

System.out.println();

}

System.out.println();

for (int v : toArray(array2d))

System.out.print(v);

System.out.println();

System.out.println();

\*/

int low = 0;

int high = toArray(array2d).length - 1;

/\*for (int v : quickSort(toArray(array2d), low, high))

System.out.print(v);

System.out.println();

System.out.println();\*/

for(int[] m : toMatrix(quickSort(toArray(array2d), low, high))) {

for(int i : m){

System.out.print(i + "\t");

}

System.out.println();

}

long time = System.currentTimeMillis();

quickSort(toArray(array2d), low, high);

System.out.println(System.currentTimeMillis() - time);

}

public static int[] toArray(int[][] arr){

int[] flat = new int[50 \* 50];

int ctr = 0;

for (int row = 0; row < 50; row++) {

for (int col = 0; col < 50; col++) {

flat[ctr++] = arr[row][col];

}

}

return flat;

}

public static int[][] toMatrix(int[] arr){

int [][] numbers = new int [50][50];

int m = 0;

for(int i = 0 ; i < 50 ; i++)

for(int j = 0 ; j < 50; j++)

numbers[i][j] = arr[m++];

return numbers;

}

public static int[] quickSort(int[] array, int low, int high) {

// выбрать опорный элемент

int middle = low + (high - low) / 2;

int opora = array[middle];

// разделить на подмассивы, который больше и меньше опорного элемента

int i = low, j = high;

while (i <= j) {

while (array[i] < opora) {

i++;

}

while (array[j] > opora) {

j--;

}

if (i <= j) {//меняем местами

int temp = array[i];

array[i] = array[j];

array[j] = temp;

i++;

j--;

}

}

// вызов рекурсии для сортировки левой и правой части

if (low < j)

quickSort(array, low, j);

if (high > i)

quickSort(array, i, high);

return array;

}

}

Код для сортировки обменом:

import java.util.Random;  
  
public class BubbleSort {  
 public static void main(String[] args) {  
  
 int[][] array2d = new int[50][50];  
 Random rand = new Random();  
  
 for (int i = 0; i < 50; i++)  
 for (int j = 0; j < 50; j++)  
 array2d[i][j] = rand.nextInt(1012);for(int[] m : *toMatrix*(*sort*(*toArray*(array2d)))) {  
 for(int i : m){  
 System.*out*.print(i + "\t");  
 }  
 System.*out*.println();  
 }  
  
 System.*out*.println();  
  
 long time = System.*currentTimeMillis*();  
 *sort*(*toArray*(array2d));  
 System.*out*.println(System.*currentTimeMillis*() - time);  
 }  
  
 public static int[] toArray(int[][] arr){  
 int[] flat = new int[50 \* 50];  
  
 int ctr = 0;  
 for (int row = 0; row < 50; row++) {  
 for (int col = 0; col < 50; col++) {  
 flat[ctr++] = arr[row][col];  
 }  
 }  
 return flat;  
 }  
  
 public static int[][] toMatrix(int[] arr){  
 int [][] numbers = new int [50][50];  
  
 int m = 0;  
 for(int i = 0 ; i < 50 ; i++)  
 for(int j = 0 ; j < 50; j++)  
 numbers[i][j] = arr[m++];  
  
 return numbers;  
 }  
  
 public static int[] sort(int[] array) {  
 *// i - номер прохода* for (int i = 0; i < array.length - 1; i++) {  
 *// внутренний цикл прохода* for (int j = array.length - 1; j > i; j--) {  
 if (array[j - 1] > array[j]) {  
 int tmp = array[j - 1];  
 array[j - 1] = array[j];  
 array[j] = tmp;  
 }  
 }  
 }  
  
 return array;  
 }  
}

Код для сортировки вставкой:

import java.util.Random;  
  
public class InsertionSort {  
 public static void main(String[] args) {  
  
 int[][] array2d = new int[50][50];  
 Random rand = new Random();  
  
 for (int i = 0; i < 50; i++)  
 for (int j = 0; j < 50; j++)  
 array2d[i][j] = rand.nextInt(1012);for(int[] m : *toMatrix*(*sort*(*toArray*(array2d)))) {  
 for(int i : m){  
 System.*out*.print(i + "\t");  
 }  
 System.*out*.println();  
 }  
  
 System.*out*.println();  
  
 long time = System.*currentTimeMillis*();  
 *sort*(*toArray*(array2d));  
 System.*out*.println(System.*currentTimeMillis*() - time);  
 }  
  
 public static int[] toArray(int[][] arr){  
 int[] flat = new int[50 \* 50];  
  
 int ctr = 0;  
 for (int row = 0; row < 50; row++) {  
 for (int col = 0; col < 50; col++) {  
 flat[ctr++] = arr[row][col];  
 }  
 }  
 return flat;  
 }  
  
 public static int[][] toMatrix(int[] arr){  
 int [][] numbers = new int [50][50];  
  
 int m = 0;  
 for(int i = 0 ; i < 50 ; i++)  
 for(int j = 0 ; j < 50; j++)  
 numbers[i][j] = arr[m++];  
  
 return numbers;  
 }  
  
 public static int[] sort(int[] array) {  
 int key;  
 for (int i = 1; i < array.length; i++) {  
 key = array[i];  
 int j = i - 1;  
 while (j >= 0 && array[j] > key) {  
 array[j + 1] = array[j];  
 j = j - 1;  
 }  
 array[j + 1] = key;  
 }  
  
 return array;  
 }  
}

Код для сортировки выбором:

import java.util.Random;  
  
public class SelectionSort {  
 public static void main(String[] args) {  
  
 int[][] array2d = new int[50][50];  
 Random rand = new Random();  
  
 for (int i = 0; i < 50; i++)  
 for (int j = 0; j < 50; j++)  
 array2d[i][j] = rand.nextInt(1012);for(int[] m : *toMatrix*(*sort*(*toArray*(array2d)))) {  
 for(int i : m){  
 System.*out*.print(i + "\t");  
 }  
 System.*out*.println();  
 }  
  
 System.*out*.println();  
  
 long time = System.*currentTimeMillis*();  
 *sort*(*toArray*(array2d));  
 System.*out*.println(System.*currentTimeMillis*() - time);  
 }  
  
 public static int[] toArray(int[][] arr){  
 int[] flat = new int[50 \* 50];  
  
 int ctr = 0;  
 for (int row = 0; row < 50; row++) {  
 for (int col = 0; col < 50; col++) {  
 flat[ctr++] = arr[row][col];  
 }  
 }  
 return flat;  
 }  
  
 public static int[][] toMatrix(int[] arr){  
 int [][] numbers = new int [50][50];  
  
 int m = 0;  
 for(int i = 0 ; i < 50 ; i++)  
 for(int j = 0 ; j < 50; j++)  
 numbers[i][j] = arr[m++];  
  
 return numbers;  
 }  
  
 public static int[] sort(int[] array) {  
 for (int i = 0; i < array.length; i++) { *// i - номер текущего шага* int pos = i;  
 int min = array[i];  
 *// цикл выбора наименьшего элемента* for (int j = i + 1; j < array.length; j++) {  
 if (array[j] < min) {  
 pos = j; *// pos - индекс наименьшего элемента* min = array[j];  
 }  
 }  
 array[pos] = array[i];  
 array[i] = min; *// меняем местами наименьший с array[i]* }  
  
 return array;  
 }  
}

Код для сортировки Шелла:

import java.util.Random;  
  
public class ShellSort {  
 public static void main(String[] args) {  
  
 int[][] array2d = new int[50][50];  
 Random rand = new Random();  
  
 for (int i = 0; i < 50; i++)  
 for (int j = 0; j < 50; j++)  
 array2d[i][j] = rand.nextInt(1012);

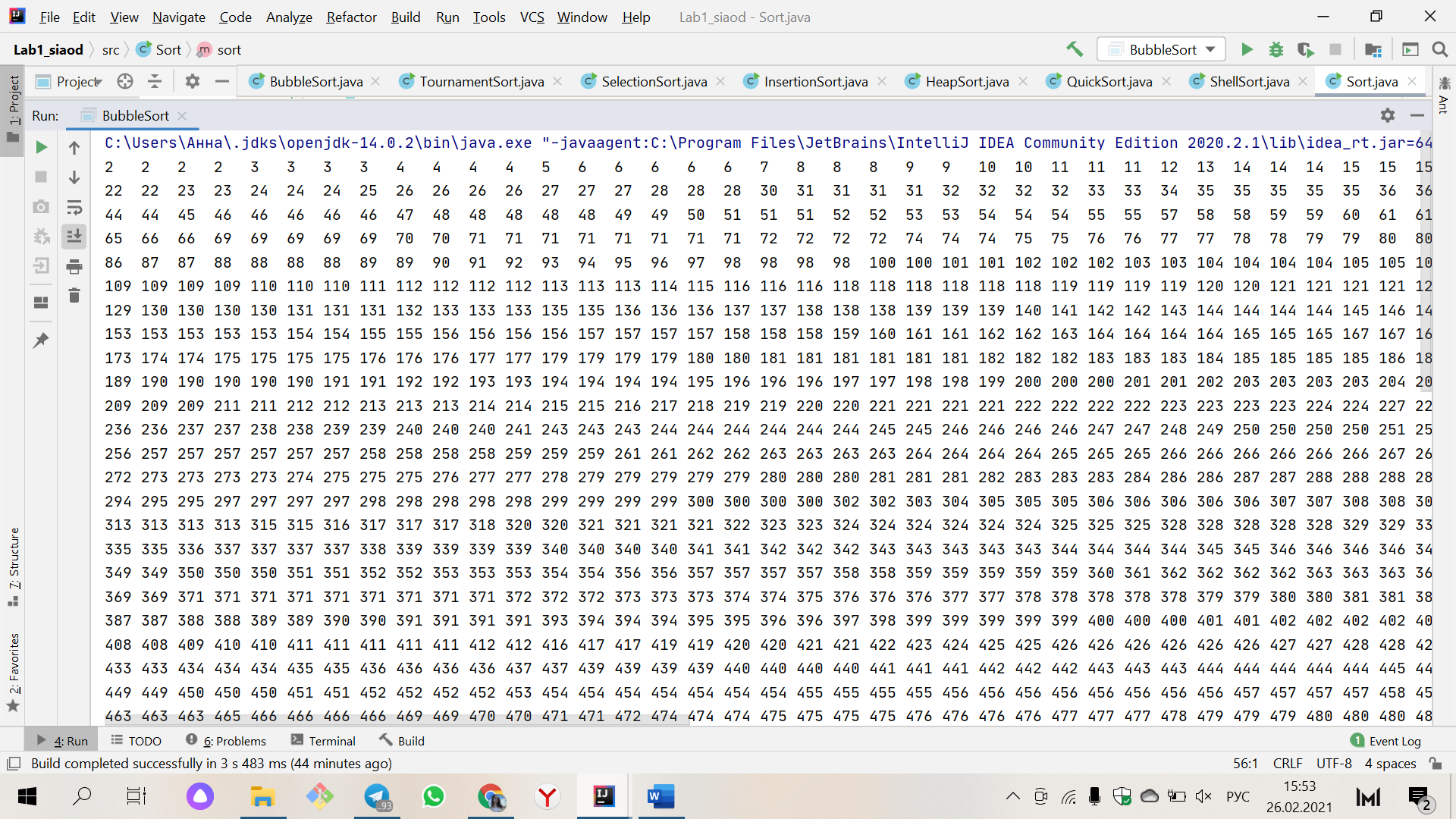
for(int[] m : *toMatrix*(*sort*(*toArray*(array2d)))) {  
 for(int i : m){  
 System.*out*.print(i + "\t");  
 }  
 System.*out*.println();  
 }  
  
 System.*out*.println();  
  
 long time = System.*currentTimeMillis*();  
 *sort*(*toArray*(array2d));  
 System.*out*.println(System.*currentTimeMillis*() - time);  
 }  
  
 public static int[] toArray(int[][] arr){  
 int[] flat = new int[50 \* 50];  
  
 int ctr = 0;  
 for (int row = 0; row < 50; row++) {  
 for (int col = 0; col < 50; col++) {  
 flat[ctr++] = arr[row][col];  
 }  
 }  
 return flat;  
 }  
  
 public static int[][] toMatrix(int[] arr){  
 int [][] numbers = new int [50][50];  
  
 int m = 0;  
 for(int i = 0 ; i < 50 ; i++)  
 for(int j = 0 ; j < 50; j++)  
 numbers[i][j] = arr[m++];  
  
 return numbers;  
 }  
  
 public static int[] sort(int[] array) {  
 int temp;  
 int h = 0;*//величина интервала  
  
 //вычисляем исходное значение интервала* while(h <= array.length/3)  
 h = 3\*h + 1;  
  
 for(int k = h; k > 0; k = (k-1)/3)  
 for(int i = k; i < array.length; i++)  
 {  
 temp = array[i];  
 int j;  
 for(j = i; j >= k; j -= k)  
 {  
 if(temp < array[j - k])  
 array[j] = array[j - k];  
 else  
 break;  
 }  
 array[j] = temp;  
 }  
  
 return array;  
 }  
  
}

Код для турнирной сортировки:

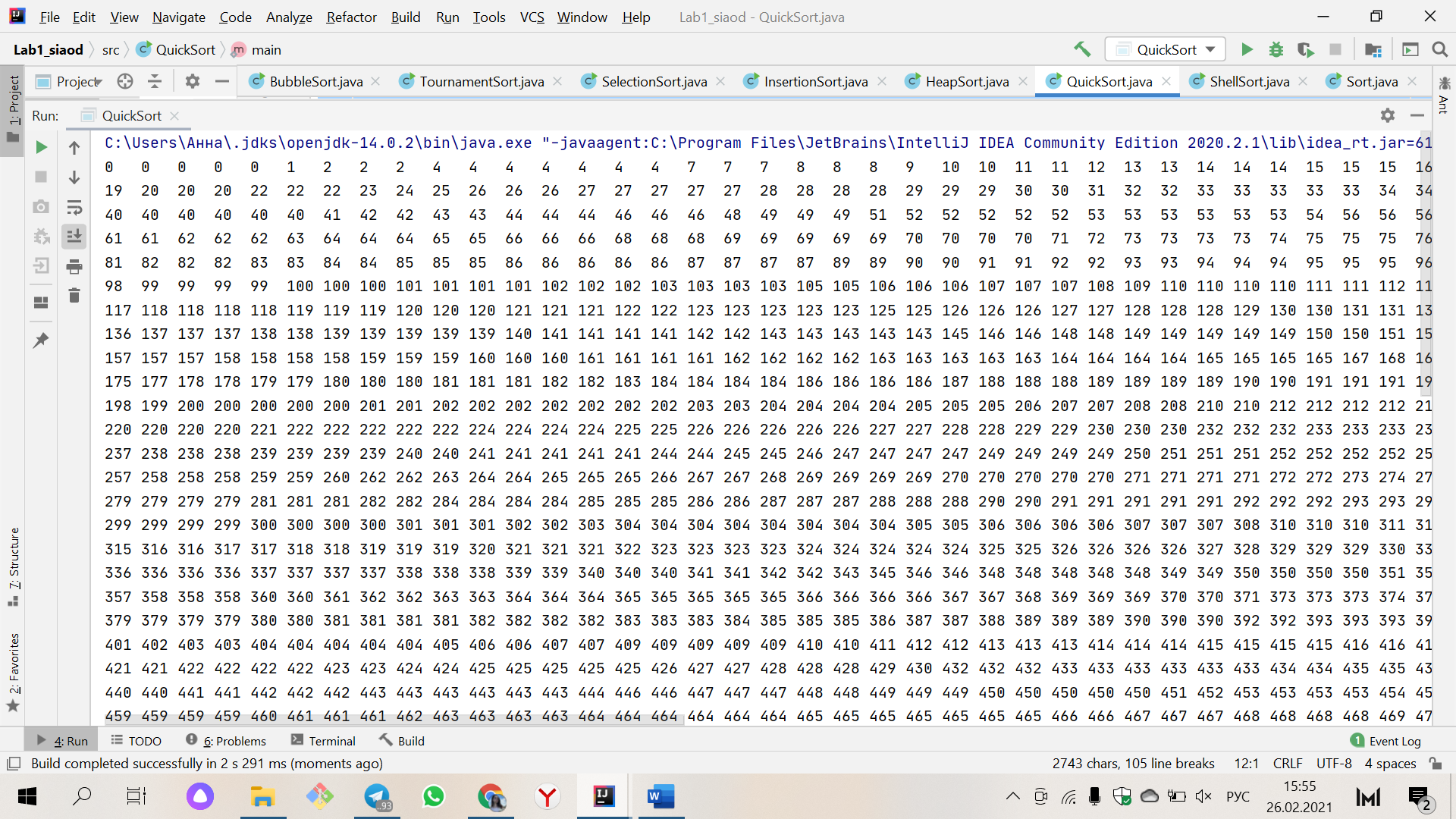
import java.util.Random;  
  
public class TournamentSort {  
 public static void main(String[] args) {  
  
 int[][] array2d = new int[50][50];  
 Random rand = new Random();  
  
 for (int i = 0; i < 50; i++)  
 for (int j = 0; j < 50; j++)  
 array2d[i][j] = rand.nextInt(1012);  
  
  
 for(int[] m : *toMatrix*(*sort*(*toArray*(array2d)))) {  
 for(int i : m){  
 System.*out*.print(i + "\t");  
 }  
 System.*out*.println();  
 }  
  
 System.*out*.println();  
  
 long time = System.*currentTimeMillis*();  
 *sort*(*toArray*(array2d));  
 System.*out*.println(System.*currentTimeMillis*() - time);  
 }  
  
 public static int[] toArray(int[][] arr){  
 int[] flat = new int[50 \* 50];  
  
 int ctr = 0;  
 for (int row = 0; row < 50; row++) {  
 for (int col = 0; col < 50; col++) {  
 flat[ctr++] = arr[row][col];  
 }  
 }  
 return flat;  
 }  
  
 public static int[][] toMatrix(int[] arr){  
 int [][] numbers = new int [50][50];  
  
 int m = 0;  
 for(int i = 0 ; i < 50 ; i++)  
 for(int j = 0 ; j < 50; j++)  
 numbers[i][j] = arr[m++];  
  
 return numbers;  
 }  
  
 public static int[] sort(int[] array) {  
 int[] arr = new int[array.length];  
 for (int i = 0; i < array.length; i++) {  
 int c;  
 for (int a = 1; a < array.length; a = a \* 2) {  
 for (int k = 0; k \* a \* 2 < array.length; k++) {  
 if (k \* a \* 2 + a < array.length) {  
 if (array[k \* a \* 2 + a] < array[k \* a \* 2]) {  
 c = array[k \* 2 \* a + a];  
 array[k \* 2 \* a + a] = array[k \* 2 \* a];  
 array[k \* 2 \* a] = c;  
 }  
 }  
 }  
 }  
 arr[i] = array[0];  
 array[0] = Integer.*MAX\_VALUE*;  
 }  
 return arr;  
 }  
}

**Работа программы**

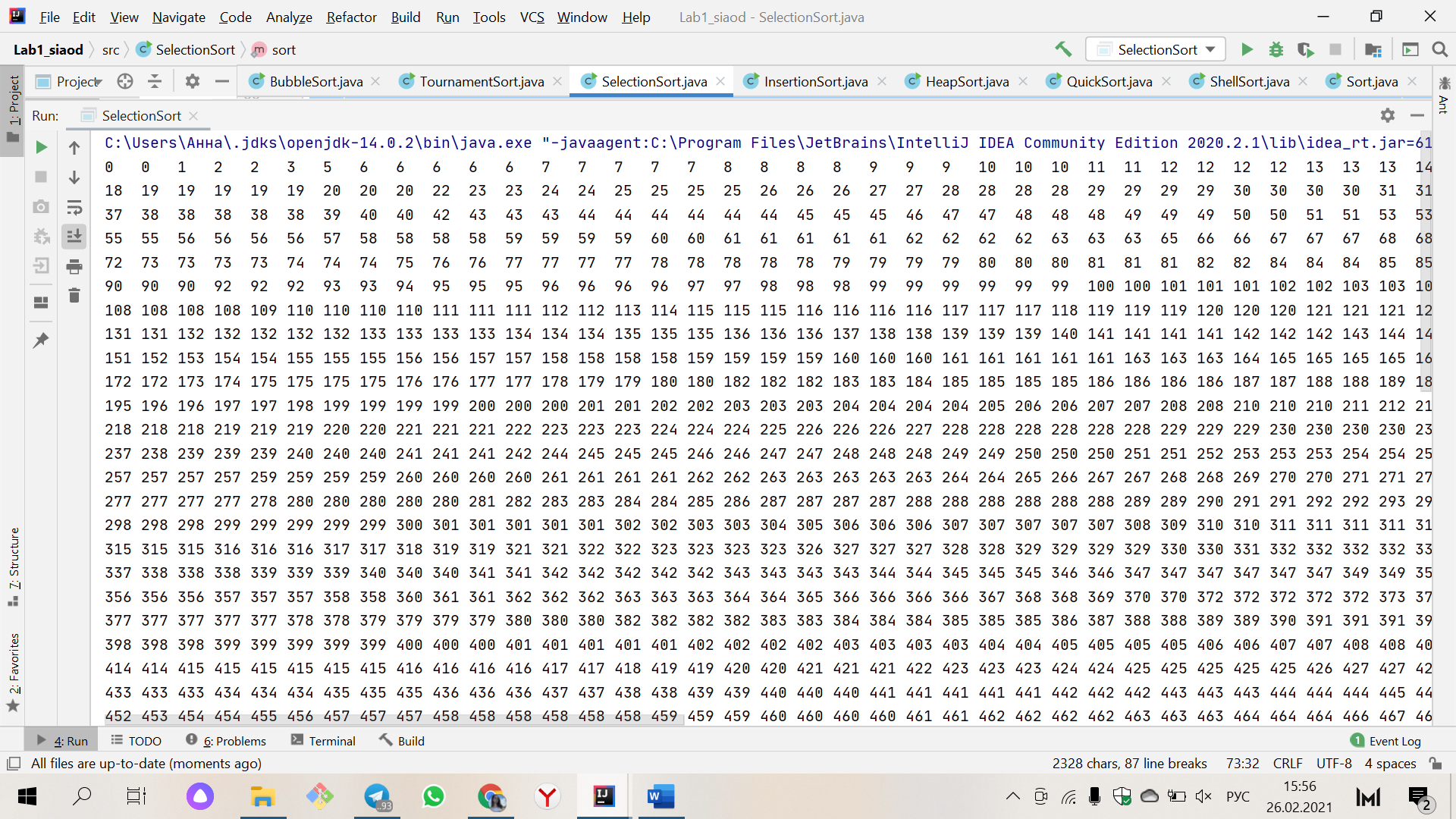
На рисунках 1-2 изображена работа программы.



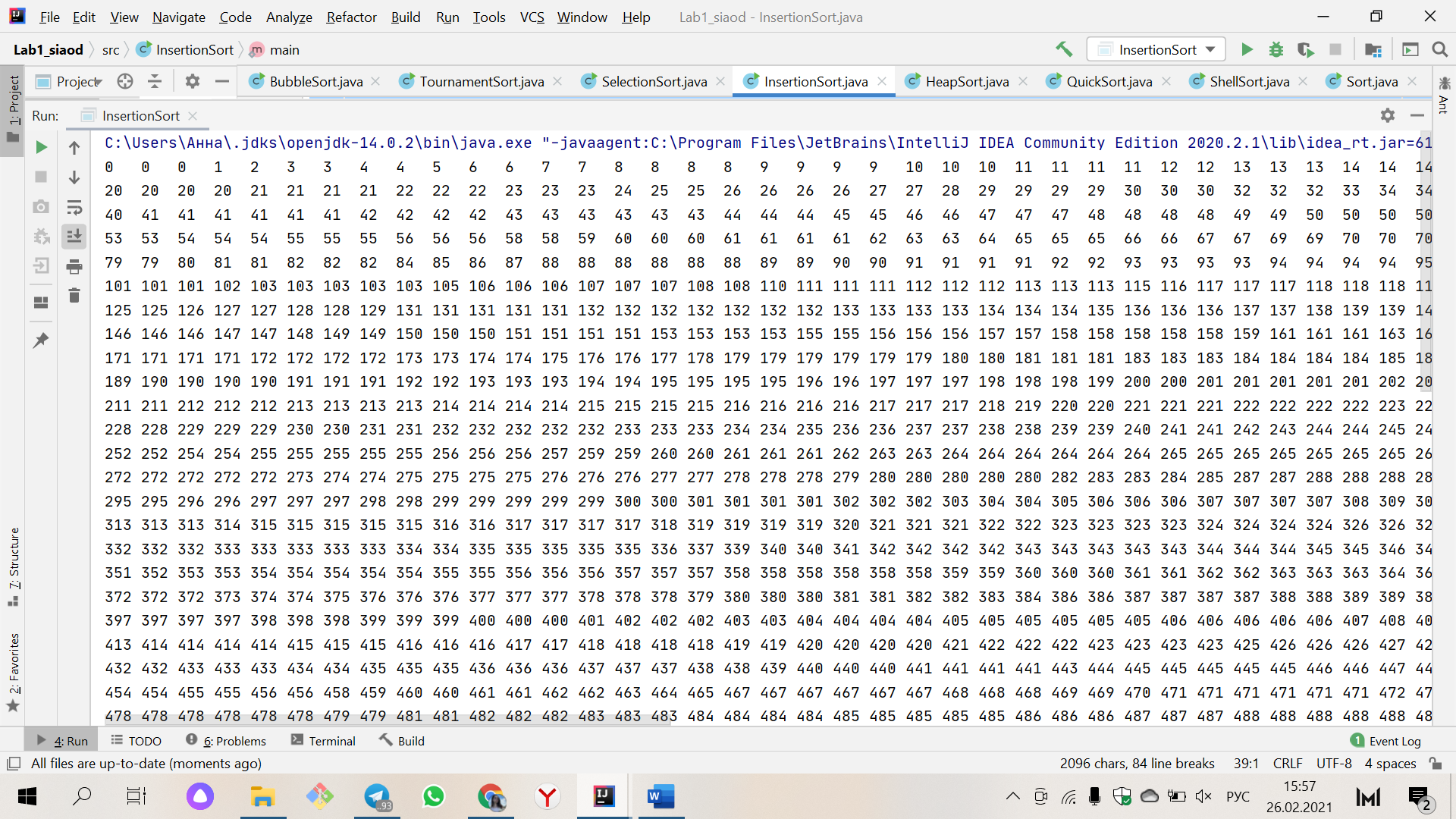
**Рисунок 1** – работа сортировки обменом.



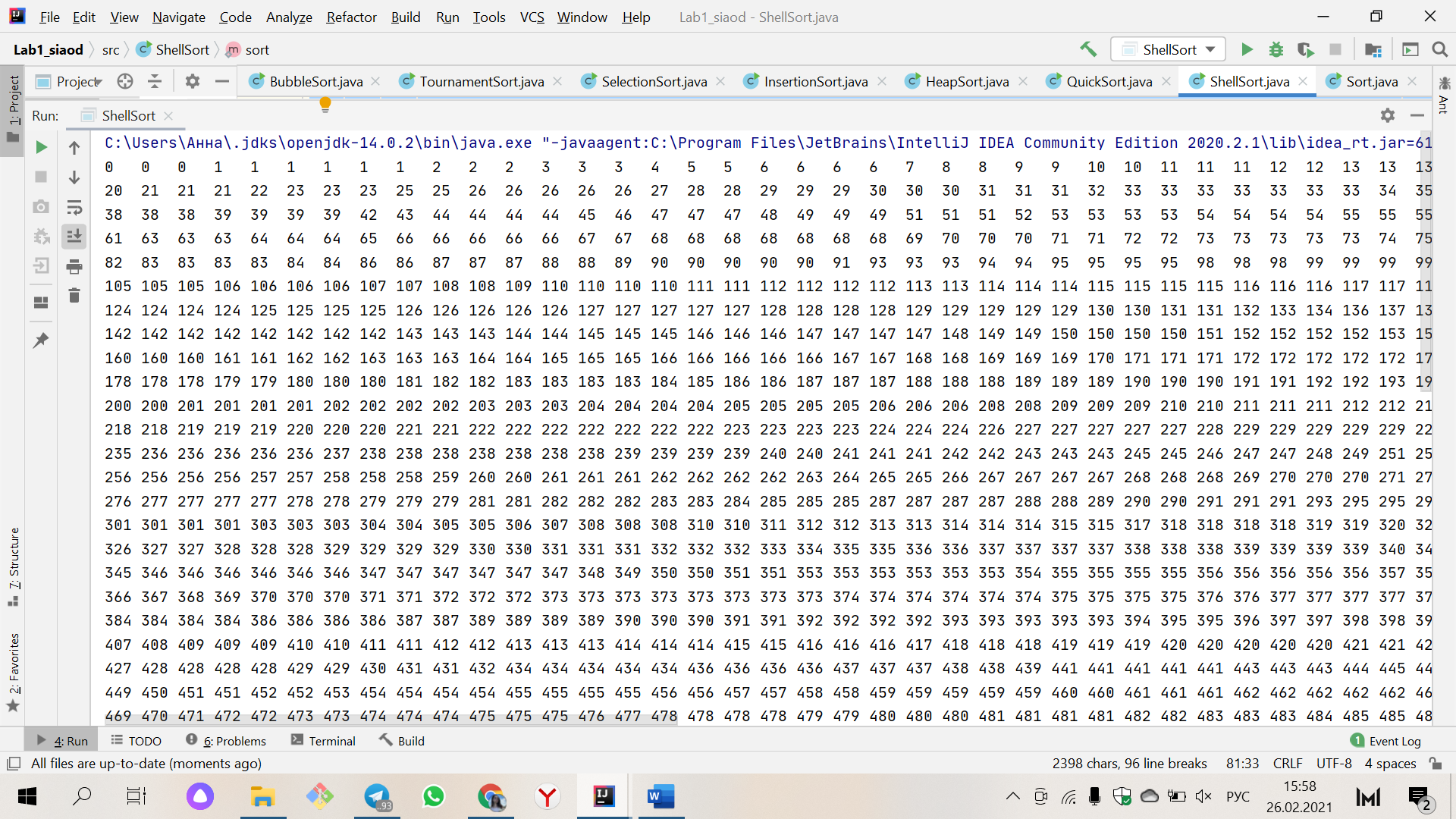
**Рисунок 2** – работа быстрой сортировки.



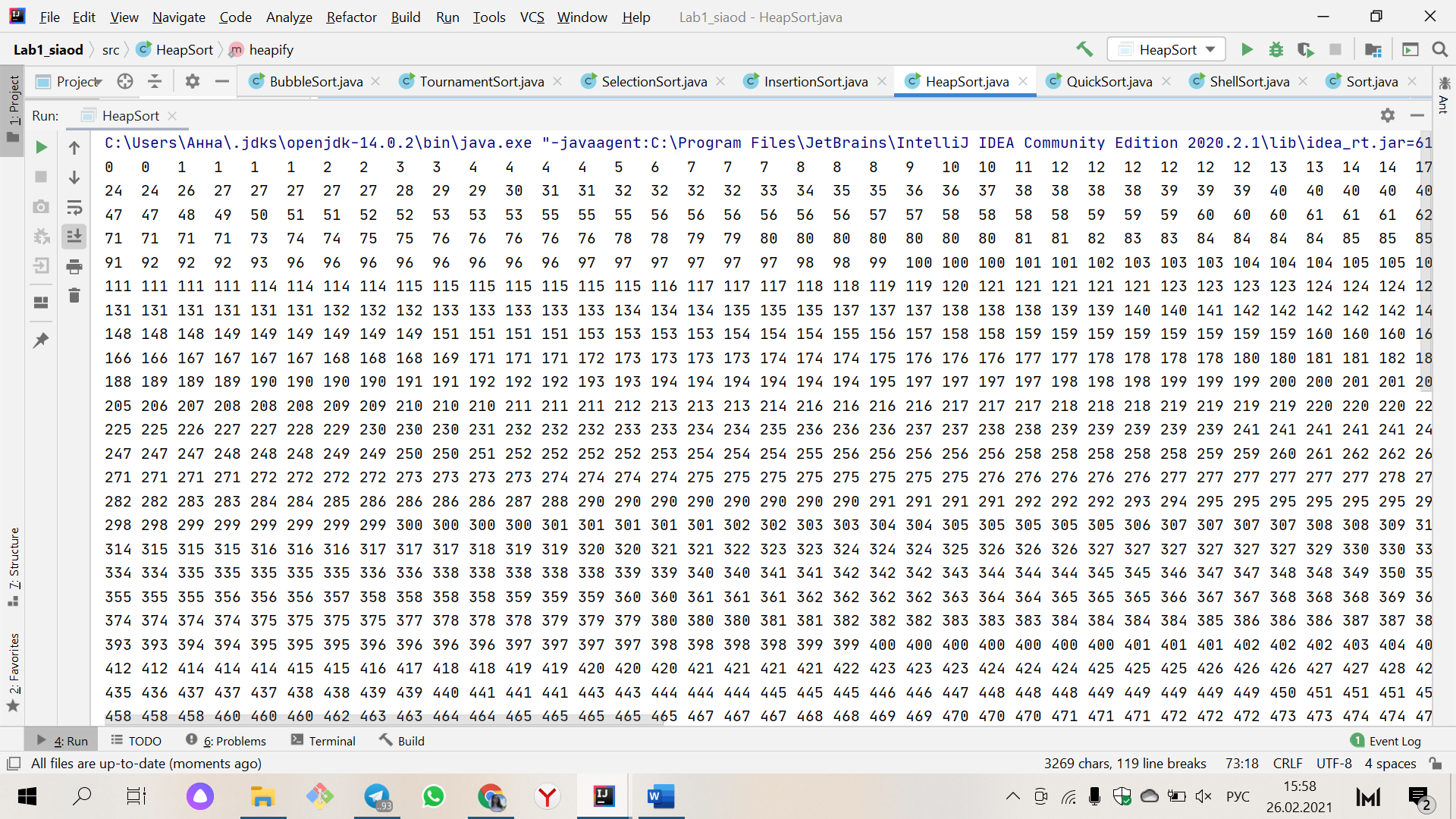
**Рисунок 3** – работа сортировки выбором.



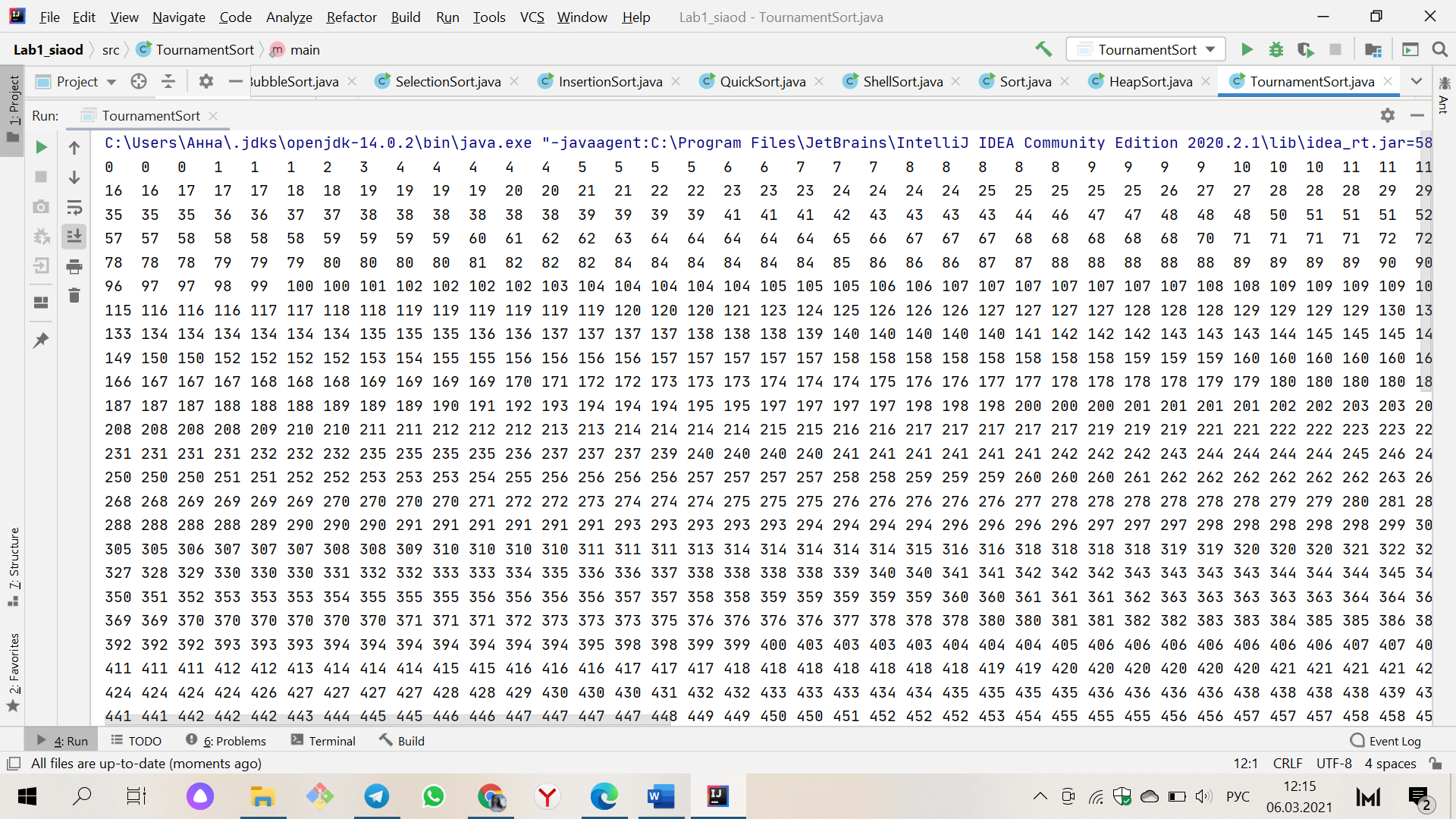
**Рисунок 4** – работа сортировки вставкой.



**Рисунок 5** – работа сортировки Шелла.



**Рисунок 5** – работа пирамидальной сортировки.



**Рисунок 6** – работа турнирной сортировки.

**Вывод:**

Реализовали методы заданных сортировок, произвели оценку времени работы каждого алгоритма.