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 СиАОД

Выполнил: студент группы БВТ1901

Адедиха Коффи Жермен

Руководитель:

Мелехин А.

Задание

Реализовать заданный метод сортировки строк числовой матрицы в соответствии с индивидуальным заданием. Для всех вариантов добавить реализацию быстрой сортировки (quicksort). Оценить время работы каждого алгоритма сортировки и сравнить его со временем стандартной функции сортировки, используемой в выбранном языке программирования. Испытания проводить на сгенерированных матрицах.

Решение:

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

Код программы сортировки выбором

```
import java.util.Random;
import java.util.Arrays;
public class Selection {
   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];
       for (int j = i + 1; j < array.length; j++) {
            if (array[j] < min) {</pre>
                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 программы сортировки вставкой

```
import java.util.Random;

public class Insertion {
    public static void main(String[] args) {
        int[][] array2d = new int[50][50];
        Random rand = new Random();

        for (int i = 0; i < 50; i++)</pre>
```

```
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++) {</pre>
        key = array[i];
        int j = i - 1;
        while (j \ge 0 \&\& array[j] > key) {
            array[j + 1] = array[j];
            j = j - 1;
        array[j + 1] = key;
```

```
return array;
}
}
```

Код программы сортировки обменом

```
import java.util.Random;
public class Bubble {
    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;
```

Код программы сортировки Шелла

```
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++) {</pre>
        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)</pre>
        h = 3*h + 1;
    for(int k = h; k > 0; k = (k-1)/3)
        for(int i = k; i < array.length; i++)</pre>
            temp = array[i];
            int j;
            for(j = i; j >= k; j -= k)
                if(temp < array[j - k])</pre>
                    array[j] = array[j - k];
                    break;
            array[j] = temp;
```

return array;

```
}
```

Код программы сортировки турнирной

```
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++) {</pre>
        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++) {</pre>
        int c;
        for (int a = 1; a < array.length; a = a * 2) {</pre>
            for (int k = 0; k * a * 2 < array.length; k++) {</pre>
                if (k * a * 2 + a < array.length) {</pre>
                     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;
```

Код программы сортироки пирамидальной

```
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++) {</pre>
           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 1 = 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 (1 < n && arr[1] > arr[largest])
           largest = 1;
       // 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;
```

Работа программы

```
1 8 16 20 20 20 33 40 47 54 58 64 76 82 89 97 99 108 116 121 128 131 141 152 159 163 171 177 224 225 224 227 228 244 227
                                                                                                                                                                                    7
16
18
25
32
39
47
54
62
75
81
87
96
98
108
101
151
158
161
158
161
159
1191
191
191
201
213
221
221
223
243
243
144
177
25
31
36
45
52
74
79
85
97
186
115
119
125
139
149
157
161
180
191
201
219
221
236
243
243
243
252
                                                          14

18

25

31

37

45

56

62

74

80

86

95

97

106

115

139

149

157

161

169

177

180

191

191

201

211

220

243

243

245

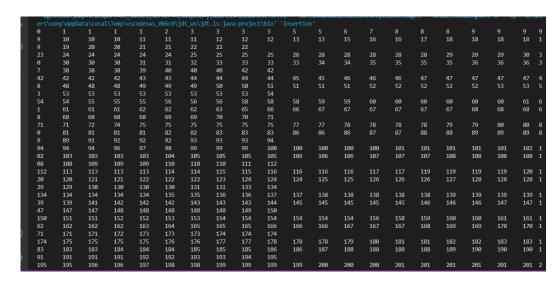
243

245

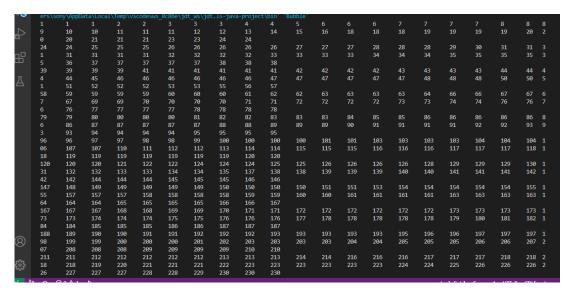
243

245
                                                                                                                                                                                                                                                                                                                                                                                                                                      21
26
36
40
47
54
58
65
77
99
108
82
117
121
129
138
141
152
160
163
172
199
204
219
224
229
238
244
257
257
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 100
108
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            100
109
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                105
114
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                124
133
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         143
154
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      143
154
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                146
156
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           185
193
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      185
194
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   186
194
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                189
196
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              204
215
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      205
216
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   206
216
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             206
216
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              230
238
```

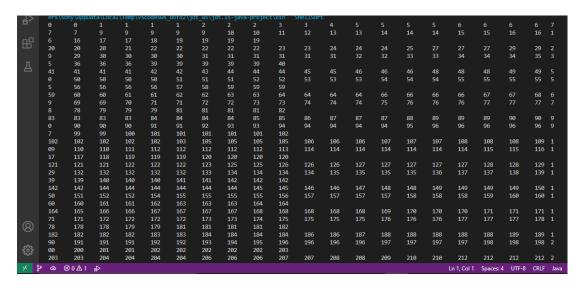
Работа сортировки Выбором.



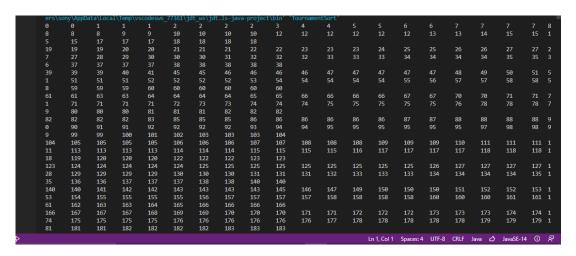
Работа вставкой сортировки.



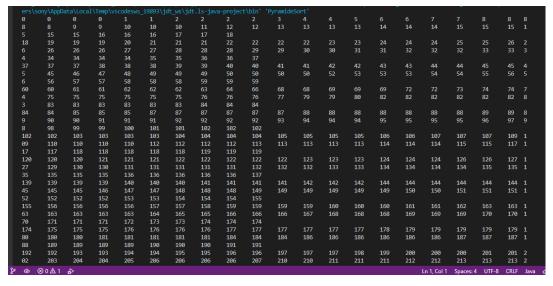
Работа сортировки обменом.



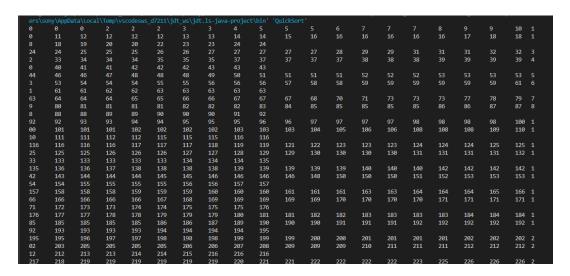
Работа сортировки Шелла.



Работа сортировки Турнирной.



Работа пирамидальной сортировки.



Работа быстрой сортировки.

Вывод:

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