

Федеральное агентство связи

**Государственное бюджетное образовательное учреждение высшего
Образование**

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Кафедра «МКиИТ»

дисциплина «СиАОД»

Отчет по Лабораторной работе №1

Подготовил студент
группы БВТ1902: Капленко Е. М.
Руководитель: Мкртчян Г. М.

Москва 2020

Задание: реализовать методы сортировки.

Методы:

Выбором	Вставкой	Обменом	Шелла	Турнирная	Быстрая сортировка	Пирамидальная
---------	----------	---------	-------	-----------	--------------------	---------------

Выполнение:

```
public static void matout(int[][] a){
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a[0].length; j++) {
            System.out.print(" " + a[i][j] + " ");
        }
        System.out.println();
    }
}

public static int[] tom(int[][] a){
    int ch = 0;
    int[] b = new int[a.length*a[0].length];

    for (int i = 0;i<a.length;i++){
        for (int j = 0;j<a[0].length; j++){
            b[ch]=a[i][j];
            ch++;
        }
    }
    return b;
}

public static List<Integer> tol(int[][] a){
    List<Integer> b = new ArrayList<Integer>();

    for (int i = 0;i<a.length;i++){
        for (int j = 0;j<a[0].length; j++){
            b.add(a[i][j]);
        }
    }
    return b;
}

public static int[][] selectionSort(int[][] a){

    int ch = 0;
    int min = Integer.MAX_VALUE;
    int k = 0;
    int ind = 0;

    long t1= System.currentTimeMillis();
    int[] b = tom(a);

    for (int i = 0;i<b.length;i++){
        for (int j = i;j<b.length; j++){
            if (min>b[j]) {min=b[j];ind = j;}
        }
        k=b[i];b[i]=min; b[ind] = k;
        min = Integer.MAX_VALUE;
    }
}
```

```

        ch = 0;
        int[][] d= new int[a.length][a[0].length];

        for (int i = 0;i<d.length;i++){
            for (int j = 0;j<d[0].length; j++){
                d[i][j]=b[ch++];
            }
        }

        long t2= System.currentTimeMillis();
        System.out.println("Selection sort = " + Long.toString((t2-t1)));
        return d;
    }

    public static int[][] sortinsert(int[][] a){
        int k;

        long t1= System.currentTimeMillis();
        int[] b=tom(a);
        for (int i = 1; i < b.length; i++) {
            k=b[i];
            for (int j = i-1; j >= 0 ; j--) {
                if(k<b[j]){
                    b[j+1]=b[j];
                    if(j==0)b[j]=k;
                }else{
                    b[j+1]=k;
                    break;
                }
            }
        }

        int ch = 0;
        int[][] d= new int[a.length][a[0].length];

        for (int i = 0;i<d.length;i++){
            for (int j = 0;j<d[0].length; j++){
                d[i][j]=b[ch];
                ch++;
            }
        }

        long t2= System.currentTimeMillis();
        System.out.println("Insertion sort = " + Long.toString(t2-t1));

        return d;
    }

    public static int[][] bubblesort(int[][] a){
        long t1= System.currentTimeMillis();
        int[] b = tom(a);
        int k=0;
        for (int i= 0;i<b.length;i++){
            for (int j = 1; j < b.length-i; j++) {

```

```

        if(b[j]<b[j-1]){k=b[j];b[j]=b[j-1];b[j-1]=k;}
    }
}
int ch = 0;

int[][] d= new int[a.length][a[0].length];

for (int i = 0;i<d.length;i++){
    for (int j = 0;j<d[0].length; j++){
        d[i][j]=b[ch];
        ch++;
    }
}

long t2= System.currentTimeMillis();
System.out.println("bubble sort = " + Long.toString(t2-t1));
return d;
}

public static int[][] Shellsort(int[][] a){
    long t1= System.currentTimeMillis();

    int[] b = tom(a);
    int k;
    int n = b.length;
    for (int step = n / 2; step > 0; step /= 2) {
        for (int i = step; i < n; i++) {
            for (int j = i - step; j >= 0 && b[j] > b[j + step] ; j -= step)
            {
                int x = b[j];
                b[j] = b[j + step];
                b[j + step] = x;
            }
        }
    }
    int ch = 0;
    int[][] d= new int[a.length][a[0].length];

    for ( int i = 0;i<d.length;i++){
        for (int j = 0;j<d[0].length; j++){
            d[i][j]=b[ch++];
        }
    }

    long t2= System.currentTimeMillis();
    System.out.println("Shell sort = " + Long.toString(t2-t1));
    return d;
}

public static int[][] Quicksort(int[][] matrix){
    long t1 = System.currentTimeMillis();
    List<Integer> list = tol(matrix);

    list = sort(list);

    int[][] result = new int[matrix.length][matrix[0].length];

    int ch = 0;

    for ( int i = 0;i < result.length; i++) {
        for (int j = 0;j < result[0].length; j++) {

```

```

        result[i][j] = list.get(ch);
        ch++;
    }
}

long t2= System.currentTimeMillis();
System.out.println("Quick sort = " + Long.toString(t2-t1));
return result;
}

private static List<Integer> concat(List<Integer> left,int p,List<Integer>
right){
    List<Integer> result = new ArrayList<Integer>();
    for (int i = 0; i < left.size(); i++) {
        result.add(left.get(i));
    }
    result.add(p);
    for (int i = 0; i < right.size(); i++) {
        result.add(right.get(i));
    }
    return result;
}

private static List<Integer> sort(List<Integer> array){
    if (array.size() < 2) return array;

    int pivot = array.get(array.size()-1);
    List<Integer> leftlist = new ArrayList<Integer>();
    List<Integer> rightlist = new ArrayList<Integer>();
    for (int i = 0;i < array.size()-1; i++) {
        if (array.get(i) > pivot) {
            rightlist.add(array.get(i));
        } else {
            leftlist.add(array.get(i));
        }
    }

    return concat(sort(leftlist),pivot,sort(rightlist));
}

private static int[][] Heapsort(int[][] matrix){
    int[] mas = tom(matrix);
    long t1 = System.currentTimeMillis();
    int n = mas.length;
    int[][] result = new int[matrix.length][matrix[0].length];

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

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

        int temp = mas[0];
        mas[0] = mas[i];
        mas[i] = temp;

        heapify(mas, i, 0);
    }
    int ch = 0;
    for ( int i = 0;i < result.length; i++) {
        for (int j = 0;j < result[0].length; j++) {
            result[i][j] = mas[ch++];
        }
    }
}

```

```

    }

    long t2= System.currentTimeMillis();
    System.out.println("Heap sort = " + Long.toString(t2-t1));
    return result;
}

private static void heapify(int[] mas, int n, int i){
    int largest = i;
    int l = 2*i + 1;
    int r = 2*i + 2;

    if (l < n && mas[l] > mas[largest])
        largest = l;

    if (r < n && mas[r] > mas[largest])
        largest = r;

    if (largest != i)
    {
        int swap = mas[i];
        mas[i] = mas[largest];
        mas[largest] = swap;

        heapify(mas, n, largest);
    }
}

private static int[][] Standard(int[][] matrix){
    long t1 = System.currentTimeMillis();
    int[][] result = new int[matrix.length][matrix[0].length];

    int[] mas = tom(matrix);
    Arrays.sort(mas);

    int ch = 0;

    for ( int i = 0; i < result.length; i++) {
        for (int j = 0; j < result[0].length; j++) {
            result[i][j] = mas[ch++];
        }
    }

    long t2= System.currentTimeMillis();
    System.out.println("Standart sort = " + Long.toString(t2-t1));
    return result;
}

```

Вывод: В ходе выполнения данной лабораторной работы реализованы методы сортировок. Ознакомилась с различными методами сортировки, узнала сложность каждой из них.