Министерство цифрового развития, связи и массовых коммуникаций

Ордена Трудового Красного Знамени федеральное государственное бюджетное

образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчёт по лабораторной работе № 1

«Методы сортировки»

по дисциплине «Структуры и алгоритмы обработки данных»

Выполнил: студент группы БВТ1902

Мартынов Николай Владимирович

Москва

2021

Оглавление

[Введение 3](#_Toc72436743)

[Листинг программы 4](#_Toc72436744)

[Вывод 10](#_Toc72436745)

Введение

Цель данной лабораторной – получить знания и навыки написав генератор случайных матриц(многомерных), который принимает опциональные параметры m, n, min\_limit, max\_limit, где m и n указывают размер матрицы, а min\_lim и max\_lim - минимальное и максимальное значение для генерируемого числа. По умолчанию при отсутствии параметров принимать следующие значения:

m = 50

n = 50

min\_limit = -250

max\_limit = 1000 + (номер своего варианта).

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.

Методы:

1. Выбором
2. Вставкой
3. Обменом
4. Шелла
5. Турнирная
6. Быстрая сортировка
7. Пирамидальная

Листинг программы

package com.company;

import java.util.\*;

//БВТ1902 Мартынов Николай 16 вариант

public class Lab\_1 {

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

if (array.length == 0)

return;//завершить выполнение, если длина массива равна 0

if (low >= high)

return;//завершить выполнение если уже нечего делить

// 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);

}

static void heapify(int[] array, int length, int i) {

int leftChild = 2\*i+1;

int rightChild = 2\*i+2;

int largest = i;

// если левый дочерний больше родительского

if (leftChild < length && array[leftChild] > array[largest]) {

largest = leftChild;

}

// если правый дочерний больше родительского

if (rightChild < length && array[rightChild] > array[largest]) {

largest = rightChild;

}

// если должна произойти замена

if (largest != i) {

int temp = array[i];

array[i] = array[largest];

array[largest] = temp;

heapify(array, length, largest);

}

}

public static void heapSort(int[] array) {

if (array.length == 0) return;

// Строим кучу

int length = array.length;

// проходим от первого без ответвлений к корню

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

heapify(array, length, i);

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

int temp = array[0];

array[0] = array[i];

array[i] = temp;

heapify(array, i, 0);

}

}

public static int[] tourSort(int[] A){

int n = 1;

int leng = A.length;

while(leng > 1){

leng = (int)Math.ceil((float)leng / 2);

n++;

}

int[][] tree = new int[n][];

leng = A.length;

for(int i = 0; i < n; i++){

if(leng != 1 && leng % 2 != 0){

leng++;

tree[i] = new int[leng];

tree[i][tree[i].length - 1] = Integer.MAX\_VALUE;

} else {

tree[i] = new int[leng];

}

leng = (int)Math.ceil((float)leng / 2);

}

System.arraycopy(A, 0, tree[0], 0, A.length);

n = Integer.MAX\_VALUE;

for(int k = 0; k < A.length; k++) {

for (int i = 0; i < tree.length - 1; i++) {

for (int j = 0; j < tree[i].length; j += 2) {

if(i == 0 && tree[i][j] == n){

tree[0][j] = Integer.MAX\_VALUE;

n = Integer.MAX\_VALUE;

}

if(i == 0 && tree[i][j + 1] == n){

tree[0][j + 1] = Integer.MAX\_VALUE;

n = Integer.MAX\_VALUE;

}

if (tree[i][j] > tree[i][j + 1]) {

tree[i + 1][j / 2] = tree[i][j + 1];

} else {

tree[i + 1][j / 2] = tree[i][j];

}

}

}

n = tree[tree.length - 1][0];

A[k] = n;

}

return A;

}

public static void main (String[]args){

// write your code here

System.out.println("Hello, world!");

int m, n, min\_limit, max\_limit;

Scanner scan = new Scanner(System.in);

System.out.println("Введите m");

m = scan.nextInt();

System.out.println("Введите n");

n = scan.nextInt();

System.out.println("Введите min\_limit");

min\_limit = scan.nextInt();

System.out.println("Введите max\_limit");

max\_limit = scan.nextInt() + 16;

// int m = 50;

// int n = 50;

// int min\_limit = -250;

// int max\_limit = 1000+16;

int[][] a = new int[m][n];

for (int i = 0; i < m; i++) {

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

// a[i][j] = min\_limit + (int) (Math.random()\*(max\_limit-min\_limit+1)); }

a[i][j] = (int) (Math.random() \* ((max\_limit - min\_limit) + 1)) + min\_limit;

}

}

System.out.println("Матрица :");

for (int i = 0; i < a.length; i++) {

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

System.out.print(a[i][j] + " ");

}

System.out.print("\n");

}

System.out.println("Сортировка Выбором :");

long time1 = System.currentTimeMillis();

for (int i = 0; i < a.length; i++) {//Перебираем массив

for (int j = 0; j < a[i].length; j++) {//Перебираем массив

for (int s = 0; s < a[i].length; s++) {//Перебираем строку

int pos = s;

int min = a[i][s];

// цикл выбора наименьшего элемента

for (int z = a[i].length - 1; z > s; z--) {//Перебираем строку

if (a[i][z] < min) {

pos = z; // pos - индекс наименьшего элемента

min = a[i][z];

}

}

a[i][pos] = a[i][s];

a[i][s] = min;

}

}

}

for (int i1 = 0; i1 < a.length; i1++) {

for (int j = 0; j < a[i1].length; j++) {

System.out.print(a[i1][j] + " ");

}

System.out.print("\n");

}

System.out.println(System.currentTimeMillis()-time1 +"ms");

System.out.println("Сортировка Вставкой :");

long time2 = System.currentTimeMillis();

for (int i = 0; i < a.length; i++) {//Перебираем массив

for (int j = 0; j < a[i].length; j++) {//Перебираем массив

for (int s = 0; s < a[i].length; s++) {//Перебираем строку

int current = a[i][s];

int z = s - 1;

while (z >= 0 && current < a[i][z]) {

a[i][z + 1] = a[i][z];

j--;

}

// выход j так же -1

// или в первом элементе, где текущий >= a[j]

a[i][z + 1] = current;

}

}

}

for (int i1 = 0; i1 < a.length; i1++) {

for (int j = 0; j < a[i1].length; j++) {

System.out.print(a[i1][j] + " ");

}

System.out.print("\n");

}

System.out.println(System.currentTimeMillis()-time2 +"ms");

System.out.println("Сортировка Обменом :");

long time3 = System.currentTimeMillis();

boolean sorted = false;

int temp;

for (int i = 0; i < a.length; i++) {//Перебираем массив

for (int j = 0; j < a[i].length; j++) {//Перебираем массив

while (!sorted) {

sorted = true;

for (int s = 0; s < a[i].length - 1; s++) {

if (a[i][s] > a[i][s + 1]) {

temp = a[i][s];

a[i][s] = a[i][s + 1];

a[i][s + 1] = temp;

sorted = false;

}

}

}

}

}

for (int i1 = 0; i1 < a.length; i1++) {

for (int j = 0; j < a[i1].length; j++) {

System.out.print(a[i1][j] + " ");

}

System.out.print("\n");

}

System.out.println(System.currentTimeMillis()-time3 +"ms");

System.out.println("Сортировка Шелла :");

long time4 = System.currentTimeMillis();

int temp1;

int h = 0;//величина интервала

//вычисляем исходное значение интервала

while (h <= a.length / 3)

h = 3 \* h + 1;

for (int i = 0; i < a.length; i++) {//Перебираем массив

for (int j = 0; j < a[i].length; j++) {//Перебираем массив

for (int k = h; k > 0; k = (k - 1) / 3)

for (int s = k; s < a[i].length; s++) {//Перебираем строку

temp1 = a[i][s];

int z;

for (z = s; z >= k; z -= k) {

if (temp1 < a[i][z - k])

a[i][z] = a[i][z - k];

else

break;

}

a[i][z] = temp1;

}

}

}

for (int i1 = 0; i1 < a.length; i1++) {

for (int j = 0; j < a[i1].length; j++) {

System.out.print(a[i1][j] + " ");

}

System.out.print("\n");

}

System.out.println(System.currentTimeMillis()-time4 +"ms");

System.out.println("Сортировка турнирная :");

long time5 = System.currentTimeMillis();

for(int i = 0; i<a.length; i++){//Перебираем массив

for(int j = 0; j<a[i].length; j++){//Перебираем массив

for (int s = 0; s<a[i].length; s++){//Перебираем строку

tourSort(a[i]);

}

}

}

for(int i1 = 0; i1<a.length; i1++){

for(int j = 0;j<a[i1].length; j++){

System.out.print(a[i1][j]+ " ");

}

System.out.print("\n");}

System.out.println(System.currentTimeMillis()-time5 +"ms");

System.out.println("Быстрая сортировка:");

long time6 = System.currentTimeMillis();

for (int i = 0; i < a.length; i++) {//Перебираем массив

for (int j = 0; j < a[i].length; j++) {//Перебираем массив

for (int s = 0; s < a[i].length; s++) {//Перебираем строку

int low = 0;

int high = a[i].length - 1;

// цикл выбора наименьшего элемента

for (int z = a[i].length - 1; z > s; z--) {//Перебираем строку

quickSort(a[i], low, high);

}

}

}

}

for (int i1 = 0; i1 < a.length; i1++) {

for (int j = 0; j < a[i1].length; j++) {

System.out.print(a[i1][j] + " ");

}

System.out.print("\n");

}

System.out.println(System.currentTimeMillis()-time6 +"ms");

System.out.println("Пирамидальная сортировка:");

long time7 = System.currentTimeMillis();

for (int i = 0; i < a.length; i++) {//Перебираем массив

for (int j = 0; j < a[i].length; j++) {//Перебираем массив

for (int s = 0; s < a[i].length; s++) {//Перебираем строку

for (int z = a[i].length - 1; z > s; z--) {//Перебираем строку

heapSort(a[i]);

}

}

}

}

for (int i1 = 0; i1 < a.length; i1++) {

for (int j = 0; j < a[i1].length; j++) {

System.out.print(a[i1][j] + " ");

}

System.out.print("\n");

}

System.out.println(System.currentTimeMillis()-time7 +"ms");

}

}

Вывод

В результате выполненной лабораторной работы я реализовал семь различных методов сортировки строк (Выбором, Вставкой, Обменом, Шелла, Турнирная, Быстрая сортировка, Пирамидальная) для сгенерированной матрицы на java.