Министерство цифрового развития, связи и массовых коммуникаций

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»



Отчет по лабораторной работе №1

по дисциплине «Структура и алгоритмы обработки данных»

по теме «Методы сортировки»

Выполнил: студент группы

БВТ1902

Бубенцов Н.В.

Проверил:

Москва

2021 г.

Оглавление

[Цель работы 3](#__RefHeading___Toc6883_2819935426)

[Задания 3](#__RefHeading___Toc6885_2819935426)

[Код программы 3](#__RefHeading___Toc6887_2819935426)

[Снимки экрана работы программы 9](#__RefHeading___Toc6889_2819935426)

[Вывод 10](#__RefHeading___Toc6891_2819935426)

## Цель работы

Разобрать принцип работы методов сортировки, используя генератор матриц с параметрами.

## Задания

1. Написать 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 = -50

max\_limit = 50

1. 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:

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

## Код программы

Программа выполнена на языке С++.  
#include<cstdlib>

#include<ctime>

#include <iostream>

#include <clocale>

using namespace std;

/// быстр сорт

void qsortRec(int\* mas, int size)

{

int i = 0;

int j = size - 1;

int mid = mas[size / 2];

do

{

while (mas[i] < mid)

{

i++;

}

while (mas[j] > mid)

{

j--;

}

if (i <= j)

{

int tmp = mas[i];

mas[i] = mas[j];

mas[j] = tmp;

i++;

j--;

}

} while (i <= j);

if (j > 0)

{

qsortRec(mas, j + 1);

}

if (i < size)

{

qsortRec(&mas[i], size - i);

}

}

//// тировка перамид

void heapif(int arr[], int n, int i)

{

int largest = i;

int l = 2 \* i + 1;

int r = 2 \* i + 2;

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

largest = l;

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

largest = r;

if (largest != i)

{

swap(arr[i], arr[largest]);

heapif(arr, n, largest);

}

}

void heapSort(int arr[], int n)

{

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

heapif(arr, n, i);

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

{

swap(arr[0], arr[i]);

heapif(arr, i, 0);

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int min = -250;

int max = 250;

int siz1 = 50;

int siz2 = 50;

int ans = 0;

srand(static\_cast<unsigned int>(time(0)));

cout << "Введите длинну строки: ";

cin >> siz1;

cout << "Введите количество строк: ";

cin >> siz2;

int\*\* arr = new int\* [siz1];

for (int i = 0; i < siz1; i++)

{

arr[i] = new int[siz2];

}

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

arr[i][j] = rand() % (max - min) + min;

}

}

cout << "Оригтнальный массив: " << endl;

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

cout << "Типы сортировки: " << endl

<< "1 - Выбором " << endl

<< "2 - Вставкой " << endl

<< "3 - Обменом " << endl

<< "4 - Шелла" << endl

<< "5 - Быстрая сортировка "

<< endl << "6 - Пирамидальная"

<< endl << "Выберети сортировку: ";

cin >> ans;

switch (ans)

{

//// Сортировка 1 типа (выбором)

case 1:

{

int min\_sort\_1 = 0;

int buf\_sort\_1 = 0;

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

min\_sort\_1 = j;

for (int k = j + 1; k < siz2; k++)

{

min\_sort\_1 = (arr[i][k] < arr[i][min\_sort\_1]) ? k : min\_sort\_1;

}

if (j != min\_sort\_1)

{

buf\_sort\_1 = arr[i][j];

arr[i][j] = arr[i][min\_sort\_1];

arr[i][min\_sort\_1] = buf\_sort\_1;

}

}

}

cout << "Сортированный массив: " << endl;

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

}

case 2:

{

//// сортировка 2 типа (вставкой)

int buf\_sort\_2;

int key\_sort\_2;

int temp\_sort\_2;

{

for (int j = 0; j < siz2; j++)

{

for (int i = 0; i < siz1 - 1; i++)

{

key\_sort\_2 = i + 1;

temp\_sort\_2 = arr[j][key\_sort\_2];

for (int k = i + 1; k > 0; k--)

{

if (temp\_sort\_2 < arr[j][k - 1])

{

arr[j][k] = arr[j][k - 1];

key\_sort\_2 = k - 1;

}

}

arr[j][key\_sort\_2] = temp\_sort\_2;

}

}

cout << "Сортированный массив: " << endl;

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

}

}

case 3:

{

//// сортировка 3 типа (обменом)

int temp\_sort\_3;

for (int k = 0; k < siz1; k++)

{

for (int i = 0; i < siz2 - 1; i++)

{

for (int j = 0; j < siz2 - i - 1; j++)

{

if (arr[k][j] > arr[k][j + 1])

{

temp\_sort\_3 = arr[k][j];

arr[k][j] = arr[k][j + 1];

arr[k][j + 1] = temp\_sort\_3;

}

}

}

}

cout << "Сортированный массив: " << endl;

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

}

case 4:

{

//// сортировка 4 типа (шелла)

int buf\_sort\_3;

int temp\_sort\_3;

for (int k = 0; k < siz1; k++)

{

buf\_sort\_3 = siz2;

buf\_sort\_3 = buf\_sort\_3 / 2;

while (buf\_sort\_3 > 0)

{

for (int i = 0; i < siz2 - buf\_sort\_3; i++)

{

int j = i;

while (j >= 0 && arr[k][j] > arr[k][j + buf\_sort\_3])

{

temp\_sort\_3 = arr[k][j];

arr[k][j] = arr[k][j + buf\_sort\_3];

arr[k][j + buf\_sort\_3] = temp\_sort\_3;

j--;

}

}

buf\_sort\_3 = buf\_sort\_3 / 2;

}

}

cout << "Сортированный массив: " << endl;

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

}

case 5:

{

//// сортировка 5 типа (быстрая сортировка)

for (int k = 0; k < siz1; k++)

{

qsortRec(arr[k], siz2);

}

cout << "Сортированный массив: " << endl;

for (int i = 0; i < siz1; i++)

{

for (int j = 0; j < siz2; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

}

}

}

## Снимки экрана работы программы

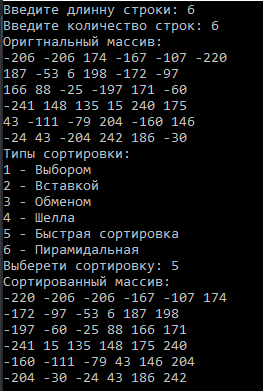
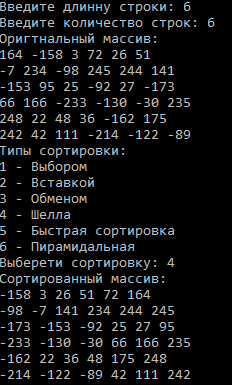
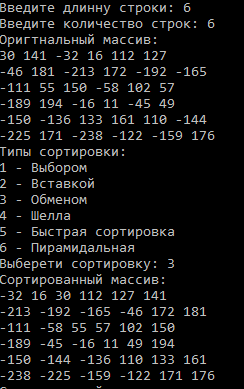
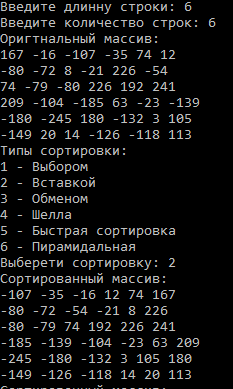
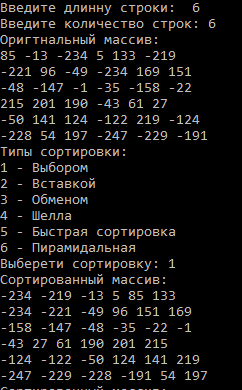


Рисунок 1 – Сортировка всех типов.

## Вывод

Я получил представление о методах сортировки данных с генерацией матриц с параметрами на языке С++.