Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (национальный исследовательский университет)»

(МАИ)

**Отчет по лабораторной работе №**1

по курсу «Программирование»

**«Оценка пузырьковой сортировки»**

|  |  |
| --- | --- |
|  | |
| Выполнил: студент группы М3О-121БВ-24 | | |
|  | Кречетов А. С. |

|  |  |
| --- | --- |
|  | |
| Проверил: преподаватель каф. 307 | |
|  | Павлов О. В. |

# Цели и задачи

В современном мире наука и технологии развиваются быстрыми темпами и часто необходимо сортировать какие-либо данные. На данный момент существует множество методов сортировки, один из которых **—** метод сортировки пузырьком.

Цель лабораторной работы: провести анализ скорости работы пузырьковой сортировки.

Задачи:

1. Сгенерировать по 10 наборов случайных данных размером 10, 100, 1000, 2000, 5000, 10000, 50000, 100000. (Значения в промежутке от -1\*10^5 до 10^5).
2. Отсортировать их пузырьковой сортировкой без оптимизаций.
3. Посчитать количество сравнений и перестановок для каждого набора.
4. Вывести данные в csv файл.

# Основная часть

Для реализации этих задач мною была написана программа на языке

программирования C++ (см. ПРИЛОЖЕНИЕ A), далее изображена блок-схема алгоритма моей программы (см. Рисунок 1).



Рисунок — Блок-схема «Оценка сортировки пузырьком»

Далее приведены результаты анализа полученных в результате работы программы данных. В начале была построена таблица со средними значениями перестановок и сравнение (см. Таблица 1).

Таблица — Среднее количество перестановок и сравнений

|  |  |  |
| --- | --- | --- |
| Количество элементов | Среднее количество перестановок | Среднее количество сравнений |
| 10 | 31 | 45 |
| 100 | 2424 | 4950 |
| 1000 | 255785 | 499500 |
| 2000 | 1035583 | 1999000 |
| 5000 | 6245705 | 12497500 |
| 10000 | 25224024 | 49995000 |
| 50000 | 625878643 | 1249975000 |
| 100000 | 2501562903 | 4999950000 |

Далее были построены графики зависимости количества перестановок от количества элементов (см. Рисунок 2) и зависимости количества сравнений от количества элементов (см. Рисунок 3).

Рисунок — Зависимости количества перестановок от количества элементов

Рисунок — Зависимости количества сравнений от количества элементов

# Заключение

В заключение можно сделать вывод, что сортировка пузырьком эффективна только на небольшом наборе данных, до 10000 элементов. Если элементов становится больше, то количества приводимых алгоритмом операций становится слишком велико и алгоритм становится не эффективным.

# ПРИЛОЖЕНИЕ

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <ctime>

void sort(int n, uint64\_t arr[], uint64\_t count\_p\_s[]){

    uint64\_t c = n-1, perest = 0, sravn = 0;

    for (int j = 0; j < n; j+=1){

        for (int i = 0; i < c; i+=1){

            sravn += 1;

            if (arr[i] > arr[i+1]){

                std::swap(arr[i], arr[i+1]);

                perest += 1;

            }

        }

    c -= 1;

    }

    count\_p\_s[0] = sravn;

    count\_p\_s[1] = perest;

}

int main(){

    int arr[8] = {10, 100, 1000, 2000, 5000, 10000, 50000, 100000};

    std::ofstream file("file.csv");

    srand(time(NULL));

    for (int j = 0; j < 8; j++){

        file << arr[j] << ';' <<"Sravnen" << ';' << "Perenos" << std::endl;

        for (int k = 0; k < 10; k+=1){

            uint64\_t count\_p\_s[2] = {};

            uint64\_t mass[arr[j]] = {};

            for (int i = 0; i < arr[j]; i++){

                mass[i] =  RAND\_MAX \* ((double)(rand())/RAND\_MAX \* (200001.0 / RAND\_MAX) - (100000.0 / RAND\_MAX));

            }

            sort(arr[j], mass, count\_p\_s);

            file << k << ":;" << count\_p\_s[0] << ';' << count\_p\_s[1];

            file << std::endl;

        }

    }

    file.close();

}