Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине: “Алгоритмы и структуры данных”

Тема: ”Изучение методов оценки алгоритмов”

Вариант №6

**Выполнил:**

Студент 1-го курса

Группы ПО-7

Комиссаров А.Е.

**Проверила:**

Глущенко Т.А.

Брест, 2020

**1) Цель работы:** Изучение методов оценки алгоритмов и программ и определение временной и ёмкостной сложности типовых алгоритмов и программ.

**Задание:** Составить программу, которая формирует матрицу из n\*n случайных чисел. Определить произведение чисел, лежащих на главной диагонали матрицы. Значение n меняется в пределах от 5 до 10 тысяч.

**2)** Алгоритм:

for (int y = 0; y < rows; ++y) {

proizv = proizv \* matrix[y][y];

}

Произведём асимптотическую и верхнюю оценку сложности заданного алгоритма по формуле:

Θ= 𝑓1+𝑓2+⋯+𝑓𝑘, где fk – трудоёмкость k-го блока.

Θ = 1 + 3\*n + n\*fцикла = 1 + 3\*n + n(1 + 1 + 2\*2) = 1 + 3\*n + 6\*n = 9\*n + 1

**3)** Исходный код программы экспериментальной оценки временной сложности заданного алгоритма для массива большой размерности:

#include <iostream>

#include <string>

#include <cmath>

#include <iomanip>

#include <time.h>

using namespace std;

int main()

{

int n; long double proizv;

cout<<"Enter N: ";

cin>>n;

clock\_t t;

t = clock();//старт таймера

proizv = 1;

/\*создание матрицы\*/

int rows = n, cols = n;

int\*\* matrix = new int\* [rows];

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

matrix[i] = new int[cols];

for (int y = 0; y < rows; ++y) {

for (int x = 0; x < cols; ++x) { //Заполнение числами от 1 до 100;

matrix[y][x] = rand() % 10 + 1;

}

}

for (int y = 0; y < rows; ++y) {

proizv = proizv \* matrix[y][y];

}

cout<<"\nProizvedenie: "<<proizv;

/\*очистка массива и его удаление\*/

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

delete[] matrix[i];

delete[] matrix;

t = clock() - t; //конец таймера

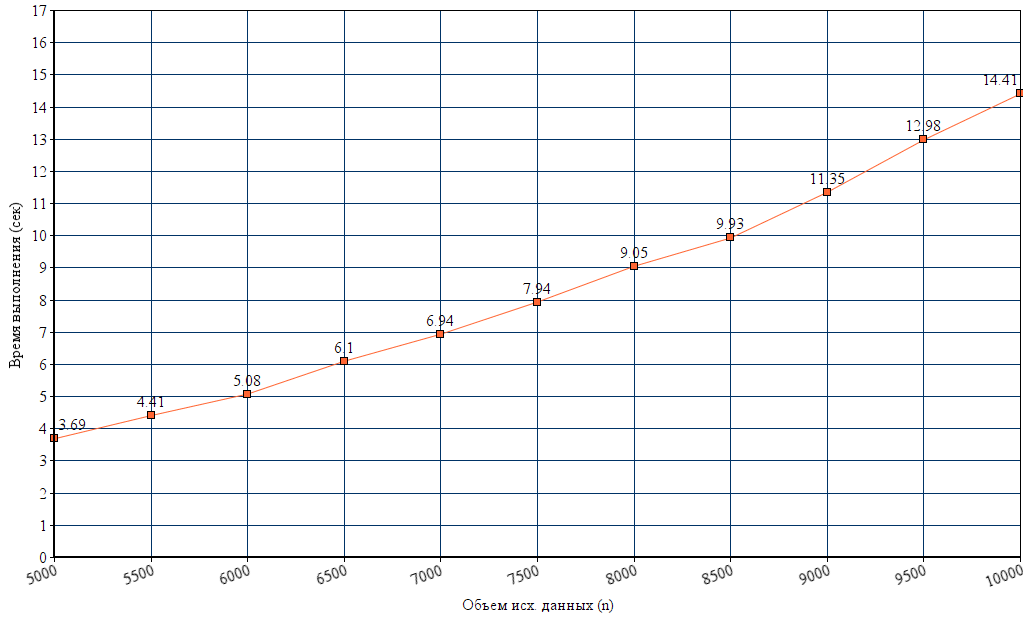
cout << "\ntime: " << ((float)t) / CLOCKS\_PER\_SEC << endl;

}

**4)**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Количество повторений | Объём исходных данных, n | Значения временной сложности, сек |
| 1 | 5 | 5000 | 3,69 |
| 2 | 5 | 6000 | 5,08 |
| 3 | 5 | 7000 | 6,94 |
| 4 | 5 | 8000 | 9,05 |
| 5 | 5 | 10000 | 14,41 |

**5)** Построим график зависимости времени выполнения алгоритма от объёма исходных данных по таблице 1:



**Вывод:** В ходе лабораторной работы изучили методы оценки алгоритмов и программ и определили временную и ёмкостную сложность типовых алгоритмов. А также с помощью графика визуально отразили зависимость временной сложности алгоритма от объёма исходных данных.