МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**КАФЕДРА ИИТ**

ОТЧЁТ

по лабораторной работе №1

**«Изучение методов оценки алгоритмов»**

Выполнил:

Студент 1 курса

группы ПО-9

Харитонович Захар Сергеевич

Проверила:

Войцехович О. Ю.

Брест 2022

**Цель работы:** Изучение методов оценки алгоритмов и программ и определение временной и емкостной сложности типовых алгоритмов и программ.

**Порядок выполнения работы**

*Вариант 10*

*Задание*

Составить программу, которая формирует матрицу из n\*n случайных чисел. Определить сумму отрицательных чисел и отдельно сумму остальных. Значение n меняется в пределах от 5 до 10 тысяч.

***Асимптотическая оценка сложности алгоритма.***

1 + 1 + 1 + 3\*n + n\*fцикла1 = 3 + 3\*n + n\*(1 + 3\*n + n\*fцикла2) = 3 + 3\*n + n\*(1 + 3n + 3n) =

= 6\*n2 + 4\*n + 3;

***Верхняя оценка сложности алгоритма.***

O(6\*n2 + 4\*n + 3) = O(n2);

***Экспериментальная оценка сложности алгоритма.***

*Код программы:*

#include <iostream>

#include <random>

#include <ctime>

using namespace std;

int main() {

srand(time(0));

int n = rand()%5001 + 5000;

cout << "n = " << n << endl;

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

for(int i = 0; i < n; i++) matrix[i] = new int[n];

for (size\_t i = 0; i < n; i++) {

for (size\_t j = 0; j < n; j++) {

matrix[i][j] = rand()%1000 - 500;

}

}

clock\_t time = clock();

long long sumP = 0, sumN = 0;

for (size\_t i = 0; i < n; i++) {

for (size\_t j = 0; j < n; j++) {

if (matrix[i][j] >= 0) {

sumP += matrix[i][j];

} else {

sumN += matrix[i][j];

}

}

}

time = clock() - time;

cout << "Time is " << time << " ms" << endl;

return 0;

}

*Результат работы:*

n = 5476

Time is 459 ms

n = 6634

Time is 598 ms

n = 7775

Time is 754 ms

n = 8258

Time is 914 ms

n = 9302

Time is 1194 ms

*График зависимости времени выполнения алгоритма от объема исходных данных.*

**Вывод:** изучены методы оценки алгоритмов и программ, выполнена асимптотическая, верхняя и экспериментальная оценки алгоритма. Результаты верхней и экспериментальной оценок алгоритма не совпадают, так как экспериментальная оценка включает в себя все операции, совершенные при выполнении программы. Верхняя оценка показывает предельную сложность, пропорциональную максимальному объёму данных.