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

Мисиюк Алексей 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 2**

*Задание*

Составить программу, которая формирует одномерный массив из n случайных чисел. Получить из него два новых массива: один из четных чисел, а другой из нечетных. Значение n меняется в пределах от 10 до 50 миллионов.

***Асимптотическая оценка сложности алгоритма.***

1 + 3\*n + n\*fцикла = 1 + 3\*n + n\*( fthen\*p+ felse\*(p-1)) = 1 + 3\*n + n\*( 5\*0.5+ 5\*0.5) = 1 + 3\*n + n\*5 =   
= 1 + 8n.

***Верхняя оценка сложности алгоритма.***

O(1 + 8n) = O(n);

***Экспериментальная оценка сложности алгоритма.***

*Код программы:*

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

int main()

{

int \*Array = new int[50000000];

int \*arrEvens = new int[50000000];

int \*arrOdds = new int[50000000];

int n, lengthEvens=0, lengthOdds=0;

cin >> n;

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

Array[i]= rand();

}

unsigned int timer\_start = clock();

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

if (!(Array[i]%2)) {

arrEvens[lengthEvens++]= Array[i];

} else {

arrOdds[lengthOdds++]= Array[i];

}

}

unsigned int timer\_stop = clock();

cout << "Time - " << timer\_stop-timer\_start << "ms" << endl;

return 0;

}

30000000

Time - 297ms

40000000

Time - 409ms

50000000

Time - 509ms

*Результат работы:*

10000000

Time - 103ms

20000000

Time - 199ms

*График зависимости* *времени выполнения(ms) алгоритма от n – объема исходных данных (кол-во элементов массива).*

**Вывод:** на практике использовали 3 метода оценки алгоритмов и программ: асимптотическая, верхняя и экспериментальная оценки. По экспериментальной оценке видно, что верхняя оценка достаточно точно дает представление о сложности алгоритма, в данном случае оценки практически совпадают, обе показывают линейную зависимость времени от количества исходных данных.