Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

По лабораторной работе №2

По курсу «Логика и основа алгоритмизации в ИЗ»

на тему «Оценка времени выполнения программ»

**Выполнил**

**студент группы 23ВВВ2:**

Федонин П.А

**Приняли:**

Митрохин М.А.

Юрова О.В.

Пенза 2024

**Цель работы:** Оценить время выполнения программы, выполняя лабораторные указания 1-2.

**Лабораторное задание 1:**

1. Вычислить порядок сложности программы (О-символику) данной программы:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main(void)

{

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

clock\_t start, end; // объявляем переменные для определения времени выполнения

int i = 0, j = 0, r;

int a[200][200], b[200][200], c[200][200], elem\_c;

srand(time(NULL)); // инициализируем параметры генератора случайных чисел

while (i < 200)

{

while (j < 200)

{

a[i][j] = rand() % 100 + 1; // заполняем массив случайными числами

j++;

}

i++;

}

srand(time(NULL)); // инициализируем параметры генератора случайных чисел

i = 0; j = 0;

while (i < 200)

{

while (j < 200)

{

b[i][j] = rand() % 100 + 1; // заполняем массив случайными числами

j++;

}

i++;

}

for (i = 0; i < 200; i++)

{

for (j = 0; j < 200; j++)

{

elem\_c = 0;

for (r = 0; r < 200; r++)

{

elem\_c = elem\_c + a[i][r] \* b[r][j];

c[i][j] = elem\_c;

}

}

}

return(0);}

Сложность данного алгоритма равен O(n3)

1. Оценить время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 100, 200, 400, 1000, 2000, 4000.

На рисунках 1-6.1 показан размер матрицы и время работы алгоритма.

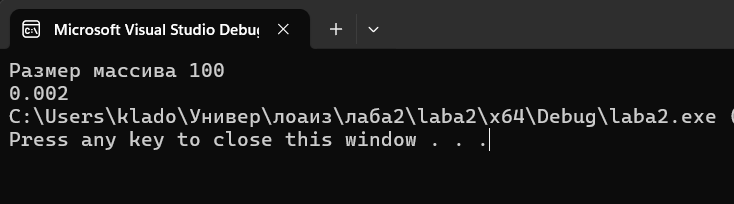


Рисунок 1

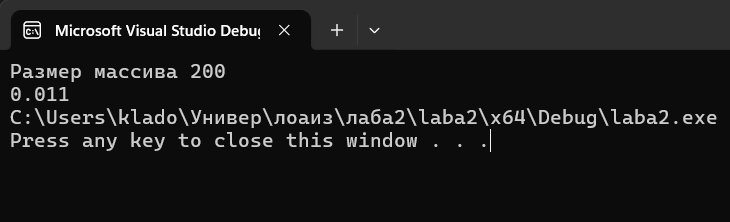


Рисунок 2

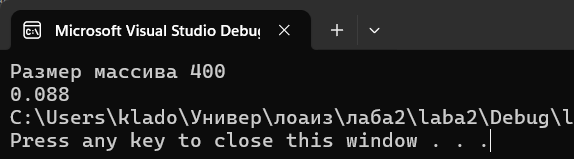


Рисунок 3

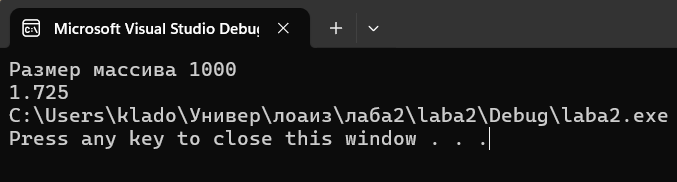


Рисунок 4

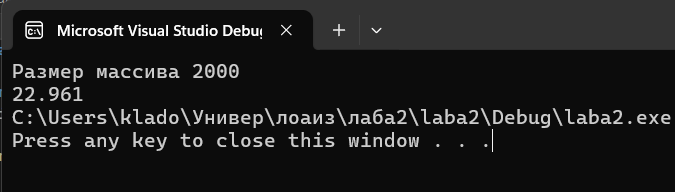


Рисунок 5

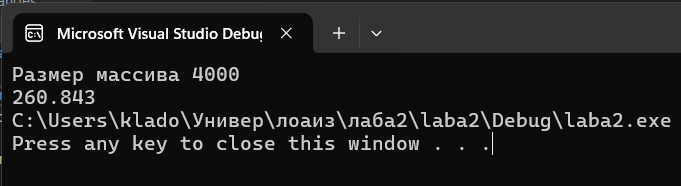


Рисунок 6

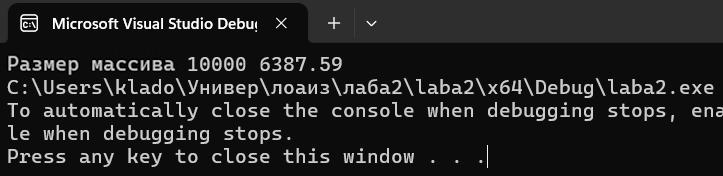


Рисунок 6.1

1. Построить график зависимости времени выполнения программы от размера матриц и сравнить полученный результат с теоретической оценкой.

На рисунке 7 показана график зависимости времени выполнения программы.

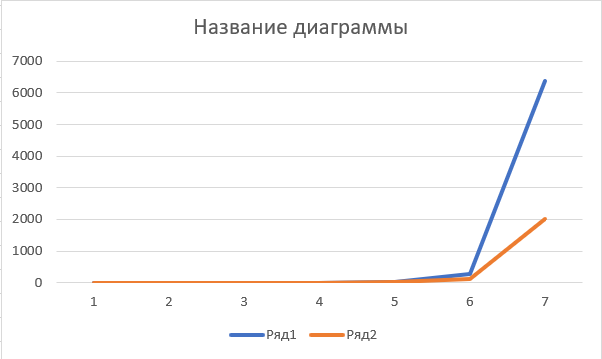


Рисунок 7

**Лабораторное задание 2:**

1. Оценить время работы каждого из реализованных алгоритмов на случайном наборе значений массива.

2. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.

3. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.

4. Оценить время работы каждого из реализованных алгоритмов на массиве, одна половина которого представляет собой возрастающую последовательность чисел, а вторая, – убывающую.

5. Оценить время работы стандартной функции qsort, реализующей алгоритм быстрой сортировки на выше указанных наборах данных.

На рисунке 8 показана таблица с временем работы алгоритмов на массиве с 100000 элементами.

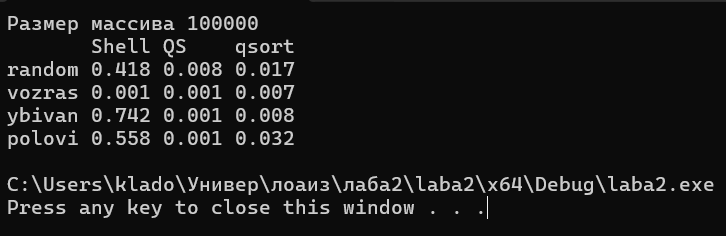


Рисунок 8

Время выполнения программы увеличивается прямо пропорционально увеличению размера массива. Qsort одинаково хорошо справляется со всеми указанными выше наборами данных.

**Вывод:** мы выполнили лабораторные указания 1-2. Оценили и сравнили между собой результаты выполнения программ.