Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

по курсу “Логика и основа алгоритмизации в ИЗ”

на тему “ Оценка времени выполнения программ”

Выполнили студенты группы 21ВВ1.3:

Лукьянова Дарья

Давкин Максим

Михальцова Полина

Приняли:

Митрохин М.А.

Юрова О.В.

Пенза 2022

**Название:**

Оценка времени выполнения программ.

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

**Лабораторное задание:**

Задание 1:

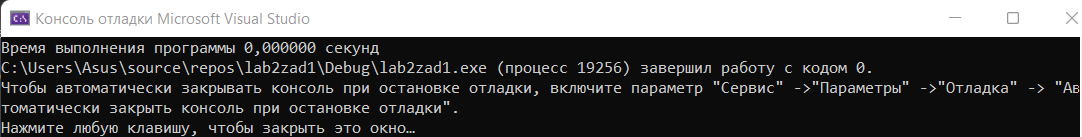
1. Вычислить порядок сложности программы (*О*-символику).
2. Оценить время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 20, 50, 75, 100, 150, 200, 250.
3. Построить график зависимости времени выполнения программы от размера матриц и сравнить полученный результат с теоретической оценкой.

Задание 2:

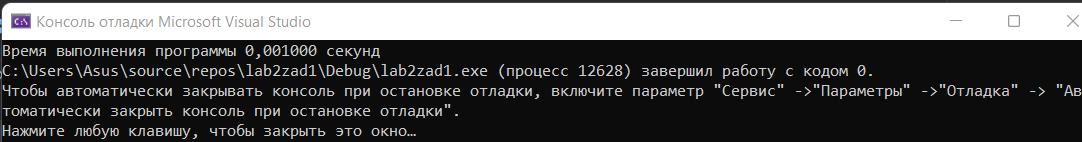
1. Оценить время работы каждого из реализованных алгоритмов на случайном наборе значений массива.
2. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.
3. Оценить время работы каждого из реализованных алгоритмов на массиве, 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.

**Задание 1:**

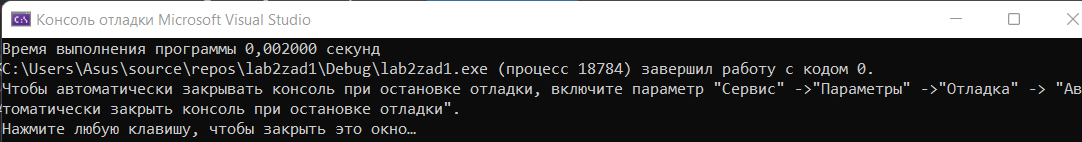
1. Рассмотрев программу, вычислили ее время выполнения О(n3).
2. Оценили время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 20, 50, 75, 100, 150, 200, 250:



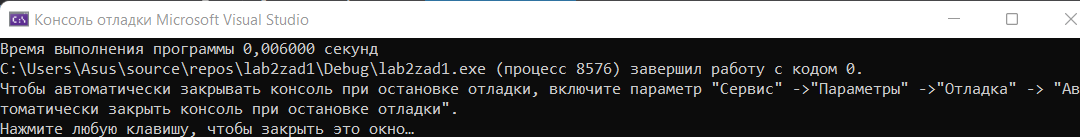
**Рис 1. Размер матрицы = 400**



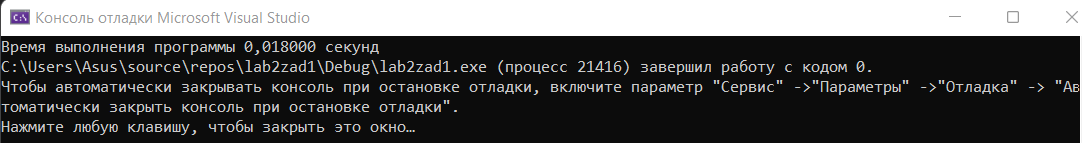
**Рис 2. Размер матрицы = 2500**



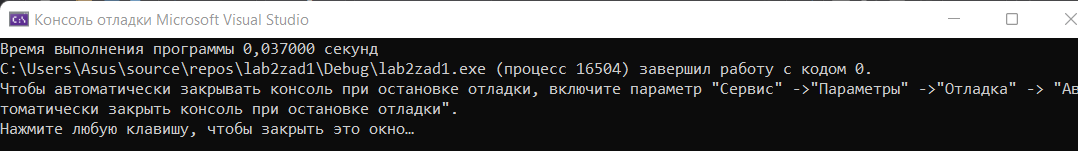
**Рис 3. Размер матрицы = 5625**



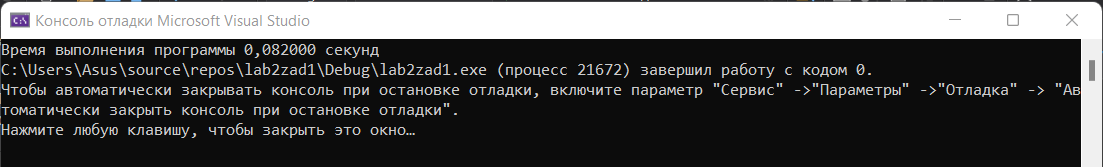
**Рис 4. Размер матрицы = 10000**



**Рис 5. Размер матрицы = 22500**



**Рис 6. Размер матрицы = 40000**



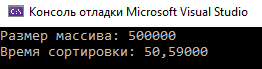
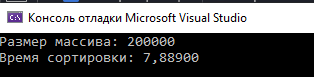
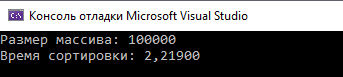
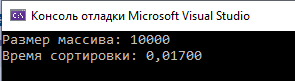
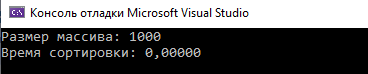
**Рис 7. Размер матрицы = 62500**

3. Построили график зависимости времени выполнения программы от размера матриц и сравнили полученный результат с теоретической оценкой.

**Задание 2:**

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

Сортировка Шелла:



Быстрая сортировка:



Видно, что время выполнения программы увеличивается прямо пропорционально размеру массива. В сравнении двух сортировок случайных чисел, быстрая сортировка справляется в разы быстрее сортировки Шелла.

1. Оценили время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.

Сортировка Шелла:



Быстрая сортировка:



Видно, что время выполнения программы (как и в п.1) увеличивается прямо пропорционально размеру массива. На возрастающей последовательности чисел быстрее сработала сортировка Шелла.

1. Оценили время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.

Сортировка Шелла:



Быстрая сортировка:

Видно, что время выполнения программы (как и в п.1) увеличивается прямо пропорционально размеру массива. На убывающей последовательности чисел быстрее сработала быстрая сортировка.



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

Сортировка Шелла:



Быстрая сортировка:

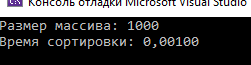
C:\Users\Максим\Downloads\Скриншот 22-09-2022 20_50_03.png



Видно, что время выполнения программы (как и в п.1) увеличивается прямо пропорционально размеру массива. На данном виде массива чисел быстрее сработала быстрая сортировка.

1. Оценили время работы стандартной функции qsort, реализующей алгоритм быстрой сортировки на выше перечисленных наборах данных.

Случайные числа:



Возрастающая последовательность:

C:\Users\Максим\Downloads\Скриншот 22-09-2022 21_08_04.png



Убывающая последовательность:



Первая половина массива возрастающая, вторая убывающая:



Видно, что время выполнения программы (как и в п.1) увеличивается прямо пропорционально размеру массива. Сортировка Qsort одинаково хорошо справляется со всеми опробованными выше наборами данных.

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