Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по Лабораторной работе №3

по курсу «Анализ Алгоритмов»

на тему: «Алгоритмы сортировки»

Студент группы ИУ7-51Б _____ (Подпись, дата) Шубенина Д. В. (Фамилия И.О.)

Преподаватель _____ (Подпись, дата) Волкова Л. Л. (Фамилия И.О.)

Преподаватель _____ (Подпись, дата) Строганов Ю. В. (Фамилия И.О.)

Москва — 2023 г.

Содержание

Введение

Сортировка является одной из самых важных операций над данными, которая имеет применения в различных задачах.

Целью данной лабораторной работы является изучение, реализация и исследование алгоритмов сортировки.

Необходимо выполнить следующие задачи:

1) изучить следующие алгоритмы сортировки:
   - классический алгоритм;
   - алгоритм Винограда;
   - оптимизированный алгоритм Винограда;
   - влгоритм Штрассена;
2) реализовать данные алгоритмы;
3) выполненить сравнительный анализ алгоритмов по затрачиваемым ресурсам (времени, памяти);
4) описать и обосновать полученные результаты в отчете.

1 Аналитическая часть

В данном разделе будут рассмотрены алгоритмы блинной, быстрой и гномьей сортировки.

1.1 Блинная сортировка

1.2 Быстрая сортировка

1.3 Гномья сортировка

Вывод

В данном разделе были рассмотрены алгоритмы умножения матриц: классический, алгоритм Винограда, алгоритм Штрассена. Для алгоритма Винограда отдельно были рассмотрены возможные оптимизации, применимые при реализации.

2 Конструкторская часть

В данном разделе будут приведены схемы алгоритмов умножения матриц, описание используемых типов данных и структуры программного обеспечения.

2.1 Требования к ПО

К программе предъявлен ряд требований:

— на вход программе подаются две матрицы, каждая записана в отдельном текстовом файле;

— результатом умножения является матрица, выводимая на экран;

— программа должна позволять производить измерения процессорного времени, затрачиваемого на выполнение реализуемых алгоритмов.

2.2 Разработка алгоритмов

2.3 Описание используемых типов данных

При реализации алгоритмов будут использованы следующие типы данных:

— *матрица* — двумерный массив значений типа `int`.

2.4 Модель вычисления для проведения оценки трудоемкости

Введем модель вычислений, которая потребуется для определения трудоемкости каждого отдельного взятого алгоритма умножения матриц.

1) Трудоемкость базовых операций имеет:

   — значение 1 для операций:

$$+, -, =, + =, - =, ==, ! =, <, >, <=, >=, \\ [], ++, --, \&\&, ||, >>, <<, \&, | \tag{2.1}$$

   — значение 2 для операций:

$$*, /, \%, * =, / =, \% = . \tag{2.2}$$

2) Трудоемкость условного оператора:

$$f_{\text{if}} = \begin{cases} \min(f_1, f_2), & \text{лучший случай} \\ \max(f_1, f_2), & \text{худший случай.} \end{cases} \tag{2.3}$$

3) Трудоемкость цикла

$$f_{\text{for}} = f_{\text{инициализация}} + f_{\text{сравнение}} + \\ + M_{\text{итераций}} \cdot (f_{\text{тело}} + f_{\text{инкремент}} + f_{\text{сравнение}}). \tag{2.4}$$

4) Трудоемкость передачи параметра в функцию и возврат из нее равен 0.

2.5 Трудоемкость алгоритмов

Вывод

3 Технологическая часть

В данном разделе приведены средства реализации программного обеспечения, сведения о модулях программы, листинг кода и функциональные тесты.

3.1 Средства реализации

В качестве языка программирования, используемого при написании данной лабораторной работы, был выбран C++ [1], так как в нем имеется контейнер `std::vector`, представляющий собой массив динамический массив данных произвольного типа, и библиотека `<ctime>` [2], позволяющая производить замеры процессорного времени.

В качестве средства написания кода была выбрана кроссплатформенная среда разработки *Visual Studio Code* за счет того, что она предоставляет функционал для проектирования, разработки и отладки ПО.

3.2 Сведения о модулях программы

Данная программа разбита на следующие модули:

— `main.cpp` — файл, содержащий точку входа в программу;

— `matrix.cpp` — файл, содержащий класс `Matrix`, реализующий необходимые для работы с матрицами функции;

— `algorithms.cpp` — файл, содержащий реализации алгоритмов умножения матриц;

— `measure.cpp` — файл, содержащий функции, замеряющие процессорное время выполнения реализуемых алгоритмов.

3.3 Реализация алгоритмов

Вывод

Были реализованы алгоритмы умножения матриц (классический, алгоритм Винограда, алгоритм Штрассена). Проведено тестирование реализованных алгортимов.

4 Исследовательская часть

4.1 Технические характеристики

Технические характеристики устройства, на котором выполнялись замеры по времени:

— Процессор: Intel i5-1035G1 (8) @ 3.600 ГГц.

— Оперативная память: 16 ГБайт.

— Операционная система: Manjaro Linux x86_64 (версия ядра Linux 5.15.131-1-MANJARO).

Во время проведения измерений времени ноутбук был подключен к сети электропитания и был нагружен только системными приложениями.

4.2 Демонстрация работы программы

4.3 Временные характеристики

4.4 Характеристики по памяти

4.5 Вывод

Заключение

В результате выполнения лаботарторной работы по исследованию алгоритмов умножения матриц решены следующие задачи:

1) описаны алгоритмы умножения матриц;
2) разработаны и реализованы соответствующие алгоритмы;
3) создан программный продукт, позволяющий протестировать реализованные алгоритмы;
4) проведен сравнительный анализ процессорного времени выполнения реализованных алгоритмов:
   - оптимизированный алгоритм Винограда оказался самым эффективным по времени независимо от размерности входных матриц;
   - время работы классического алгоритма умножения и оптимизированного и неоптимизированного алгоритмов Винограда на матрицах нечетного размера больше, чем на матрицах четного размера; для алгоритма Винограда меньшая скорость работы на матрицах нечетного размера объясняется необходимостью дополнительных вычислений крайних строк и столбцов в результирующей матрице;
   - алгоритм Штрассена показал наименьшую производительность среди всех алгоритмов, исследуемых на матрицах, размер которых равен степени двойки; низкая производительность алгоритма обуславливается необходимостью выполнения дополнительных операций сложения/вычитания, разбиения/слияния матриц;
5) выполнена теоретическая оценка объема затрачиваемой памяти каждым из реализованных алгоритмов: стандартный алгоритм умножения матриц является наименее ресурсозатратным из всех реализованных алгоритмов; самым же требовательным по памяти оказался алгоритм Штрассена за счет использования вспомогательных подматриц

для выполнения рассчетов и рекурсивных вызовов; оптимизированный алгоритм Винограда использует больше ресурсов по сравнению с неоптимизированным, за счет предвычисления некоторых выражений.

Список использованных источников

1 Документация по Microsoft C++ [Электронный ресурс]. — Режим доступа: https://learn.microsoft.com/ru-ru/cpp/?view=msvc-170&viewFallbackFrom=vs-2017 (дата обращения: 25.09.2023).

2 Standard library header <ctime> [Электронный ресурс]. — Режим доступа: https://en.cppreference.com/w/cpp/header/ctime.