ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 5

«Сортировки»

Выполнил работу

Максудов Кирилл

Академическая группа J3111

Принято

Практик Вершинин Владислав

Санкт-Петербург

2024

1. Введение

Цель работы: написать и проанализировать 3 алгоритма сортировки.

Задачи:

1) Написать алгоритм

2) Измерить время работы алгоритма

3) Сравнить алгоритмы на разных данных

1. Теоретическая подготовка

Использовал векторы, while , for циклы, различные типы данных.

1. Реализация:

Bubble sort: проходим по вектору в двух вложенных циклах, конец второго цикла постоянно сдвигается на 1 элемент влево, по время прохода постоянно сдвигаем самый большой элемент вправо.

Quick sort: берем средний элемент вектора, ищем элемент которые меньше среднего и который больше с концов вектора, меняем их местами. Рекурсивно повторяем все предыдущие действия для левой и правой части.

Counting sort: ищем максимальный элемент вектора. Создаем два вектора длинны сортируемого вектора(далее output) и максимального значения(далее count) и заполняем их нулями. Считаем количество каждого уникального значения, в цикле увеличивая значение в ячейках с соответствующим данному значению индексом в count. Создаем кумулятивную сумму проходя по count. Ищем индексы каждого эл исходного вектора в count. Копируем отсортированное в исходный вектор.

1. Экспериментальная часть

Подсчет по памяти:

1)Bubble sort: исходный вектор, счетчики в цикле, длинна

2)Quick sort: std::исходный вектор

3)Counting sort: int output[n] – 32\*N бит

int count[k] – 32\*N бит, где k – макс. Эл. вектора

Подсчет асимптотики:

1)Bubble sort: два вложенных цикла - O(N^2) - всегда

2)Quick sort: в худшем случае – O(N^2)

3)Counting sort: несколько невложенных циклов по вектору – O(N\*k) k -значительно меньше N

Таблица №1 – Подсчет сложности реализованных алгоритмов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Количество элементов | 50 | 100 | 500 | 1000 |
| Bubble sort | 0.00000940 | 0.00003420 | 0.00280920 | 0.00500660 |
| Quick sort | 0.00000570 | 0.00001400 | 0.00006090 | 0.00006950 |
| Counting sort | 0.00001220 | 0.00000770 | 0.00001470 | 0.00003360 |

Изображение №1 – График зависимости времени от числа элементов

\*значение для Bubble sort с 500 эл. были изменены, чтобы было лучше видно остальные сортировки, на самом деле там гораздо больше времени

Изображение выглядит как линия, снимок экрана, диаграмма, График

Автоматически созданное описание

Изображение выглядит как линия, снимок экрана, диаграмма, График

Автоматически созданное описание

Изображение №2 – Box plot для quick b counting сортировок на 10e5

Изображение выглядит как линия, диаграмма, снимок экрана, График

Автоматически созданное описание

Изображение выглядит как линия, снимок экрана, диаграмма, График

Автоматически созданное описание

Изображение №3 – Box plot для quick b counting сортировок на 10e6

1. Вывод:

В асимптотике мы описываем худший случай из за чего на графике реальное время выглядит меньше. Выбросы есть на худших и лучших случаях для конкретного алгоритма. Алгоритм хорошо работают почти всегда, но Couting работает быстрей и тратит меньше памяти при меньших числах в списке, а Quick работает быстрей при “случайном” .

1. Заключение

В ходе работы мною было реализовано 3 алгоритма сортировки. Из них эффективными оказались Quick и Counting, на большем количестве элементов Counting оказался быстрей, однако он требует немного больше памяти

ПРИЛОЖЕНИЕ А

Листинг кода файла lab-5.cpp

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание