

Лабораторная работа № 1 по курсу дискретного анализа: сортировка за линейное время

Выполнила студентка группы 08-207 МАИ *Михеева Кристина*.

Условие

Кратко описывается задача:

1. Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.
2. Вариант 2-3. Метод сортировки: Сортировка подсчётом. Тип ключа: почтовые индексы. Тип значения: числа от 0 до 264 - 1.

Метод решения

В структуру записываем значение ключей и значение индексов. Далее осуществляем сортировку подсчетом: создаем вектор на 100000 элементов, заполненными нулями, затем проходимся по всем значениям вектора и инкрементируем, потом считаем префиксные суммы, создаем вспомогательный вектор на размер исходного, чтобы там были отсортированные данные, идем с конца вектора и в отсортированный вектор в текущий индекс пишем текущее значение. После всего, считывание элементов в контейнер, потом его сортировка и потом вывод элементов.

Общее описание алгоритма решения задачи.

Имеется один главный файл lab1.cpp, где есть структура для, включающая в себя два поля - key и value набор таких структур необходимо реализовать и сама реализация сортировки подсчета.

Дневник отладки

При запуске задания в констексте, возникли некоторые проблемы, которая была быстро устранена. Не подключила include <iomanip>, данная библиотека используется для модификации форматов вывода сначала. Эту проблему смогла решить, воспользовавшись доп. информацией, используя источники из Интернета.

Тест производительности

Тест производительности:

Для 10000 тестов соответственно 0,074 секунд

Для 100000 тестов соответственно 0,737 секунд

Для 1000000 тестов соответственно 7,277 секунд

Делая вывод, мы можем заметить, что сложность примерно линейная.

Недочёты

В данной лабораторной работе недочетов не обнаружено.

Выводы

В данной лабораторной работе я познакомилась с типом сортировки, который имеет название сортировка подсчетом. Данная сортировка имеет сложность $O(n+k)$, где n - длина массива, а k - максимальный элемент из исходного массива. По моему мнению, данная сортировка является тривиальной. Сортировка подсчетом лучше всего работает, когда у нас достаточно большое количество одинаковых целых значений. Также было изучено много разных способов для оптимизации работы по памяти и по времени, что является очень важным результатом для дальнейших работ.