

Лабораторная работы № 1 по курсу дискретного анализа:

Сортировки за линейное время

Выполнил студент группы 08-210 МАИ *Некрасов Константин*

Условие

Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.

- Вариант задания: сортировка подсчётом
- Тип ключа: числа от 0 до 65535.
- Тип значения: строки фиксированной длины 64 символа, во входных данных могут встретиться строки меньшей длины, при этом строка дополняется до 64-х нулевыми символами, которые не выводятся на экран.

Метод решения

Исходные пары входных данных записываются в отдельные массивы под ключи и под значения. Чтобы не нарушить связи в парах, действия с индексами в массиве ключей также должны влиять на индексы в массиве значений.

Помимо массивов ключей и значений также нужен массив, в котором будет отображено, сколько раз встречается каждый ключ вплоть до максимального. После окончания подсчёта, в полученном массиве к каждому его элементу начиная с первого нужно прибавить значение предыдущего, чтобы учесть количество меньших значений, а не только равных.

Дальше заполним массив ключей начиная с конца соответствующими значениями и соответственно заполним массив значений, и на выходе получим отсортированные массивы, которые нужно правильно вывести в формате «ключ-значение».

Описание программ

По дополнительному условию, при реализации сортировки подсчётом запрещено использовать `std::vector`, поэтому для реализации динамического массива использовались указатели. Изначальный размер для массивов ключей и значений равен 10000, но может возникнуть необходимость в большем размере, поэтому была сделана функция `Resize`, которая создаёт

новые массивы ключей и значений большего размера, копирует в них данные, после чего очищает исходные массивы и присваивает им новые.

В саму функцию сортировки подсчётом **CountingSort** передаются не только исходные массивы данных, но и массивы, в которые они будут записаны, а также массив, в который будут записано количество «встреч» (память под все эти массивы заранее выделена). Внутри функции выполняются все те действия, которые описаны в разделе **Метод решения**.

Для считывания строки из потока ввода использовались функции **getline** и **strncpy** с выделением памяти при необходимости для правильного считывания данных и вписывания их в массив значений.

Для вывода отсортированных массивов использовался цикл, в котором для каждого *i*-ого ключа выводилось *i*-ое значение. После вывода, вся выделенная память была очищена.

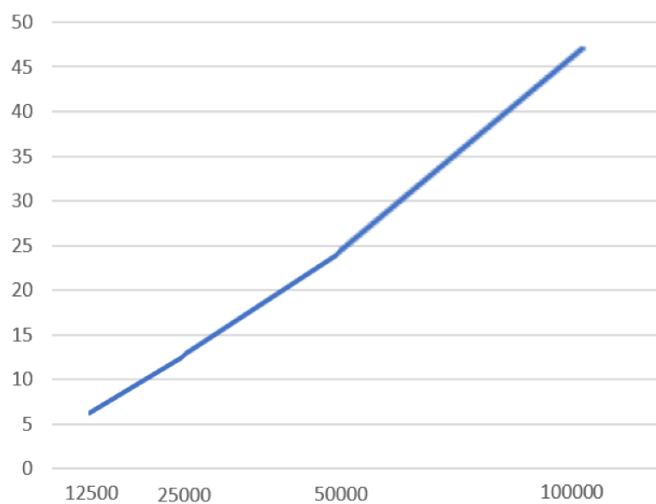
Дневник отладки

Программа прошла все тесты системы с первой попытки.

Тест производительности

Ось Oy: время в 10^{-4} с

Ось Ox: объём данных *n*



Согласно графику, линейность сортировки сохраняется.

Недочёты

Явных недочётов не было обнаружено.

Выводы

При написании программы был реализован алгоритм сортировки за линейное время, который может быть эффективно применён для массивов с часто повторяющимися значениями.