

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

Выполнил студент группы 08-208 МАИ *Сикорский Александр*.

Условие

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 Тип ключа: почтовые индексы. Тип значения: числа от 0 до $2^{64} - 1$.

Метод решения

Для написания алгоритма было достаточно послушать лекцию. Нам была нужна устойчивая сортировка подсчетом, то есть та, которая не переставляет элементы с равными ключами. Для этого нужно было дописать метод, который насчитывает префикс-суммы в векторе, содержащем количество элементов со значением n . Получили вектор с элементами следующего смысла: **значение элемента с индексом i указывает, сколько имеется элементов $\leq i$ в исходном векторе..** Это без учета нижней границы содержащихся элементов. После прошли с конца исходного вектора и в зависимости от значений вектора префикс-сумм сформировали результат сортировки.

Описание программы

В `main.cpp` написан ввод, определение левой и правой границы полученных ключей, вызов сортировки и форматированный вывод результата. В `TKeyValue.h` лежит класс сортируемых элементов (ключ + значение), а в `TCountingSort.h` метод сортировки подсчетом и приватные методы, которые вызываются из него.

Дневник отладки

От посылки к посылке я код не менял, первый раз заслал архив с мейком, хотя на я.контакте был выбран другой компилятор, потом после перезагрузки страницы не посмотрел и заслал архив на другую задачу, с первой корректной посылки код ОК'нулся. Во время локального тестирования пришлось погуглить про форматирование вывода, еще не с первого раза грамотно написал алгоритм в плане зависимости порядка элемента от индекса в векторе префикс-сумм.

Тест производительности

1. 100 эл. - 0.023s

2. 10_000 эл. - 0.527s
3. 100_000 эл. - 7.205s
4. 1_000_000 эл. - 1m19.641s

Недочёты

В первой и единственной корректной посылке на чекере есть такая строчка: `std::setfill('0')`. Мне кажется это попадает под запрет на использование неименованных констант. За-метил и исправил это я уже после того, как отправил архив на тесты.

Выводы

Сортировка подсчетом может использоваться эффективно, когда диапазон ключей сильно меньше количества сортируемых элементов. Пусть n, k - размер входных данных и диапазон ключей соответственно. Сначала нужно определить верхнюю и нижнюю границы диапазона поступивших на сортировку ключей. Делаем это за $\Theta(n)$, затем за $\Theta(n)$ же заполняем новый вектор размера k , в котором насчитано количество ключей с разными значениями. Далее за $\Theta(k)$ пройдем по полученному вектору и посчитаем в нем префикс-суммы для каждого элемента. Формирование ответа занимает $\Theta(n)$, тут мы идем по исходному вектору с конца. Имеем временную сложность $\Theta(n + k)$ и пространственную $\Theta(n + k)$. Описание проблем, возникших при решении задач есть в разделах "Недочеты" и "Дневник отладки".