

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

Выполнил студент группы 08-208 МАИ *Попов Матвей*.

## 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 задания: Сортировка подсчётом.
3. Тип ключа: числа от 0 до 65535.
4. 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

Для сортировки подсчётом исходного массива пар «ключ - значение» создадим два вспомогательных массива: `count` для счётчика и `ans` для отсортированного массива. Заполняем массив `count` нулями, затем увеличиваем `count[raw[i]]` на 1 для каждого `raw[i]`. Затем к каждому `count[i]`, начиная с `count[1]`, прибавим `count[i - 1]`, чтобы знать количество меньших или равных элементов для каждого элемента исходного массива. Наконец прочитаем с конца (для устойчивости) исходный массив, значение `count[raw[i]]` уменьшим на 1 и в `ans[count[raw[i]]]` запишем `raw[i]`.

## Описание программы

Из-за проблем с чекером вся программа была переписана в один файл `lab.cpp`, был реализован класс `TPair` для хранения пар «ключ-значение», функция `Count` для заполнения массива `count` и функция `Sort` для заполнения массива `ans`.

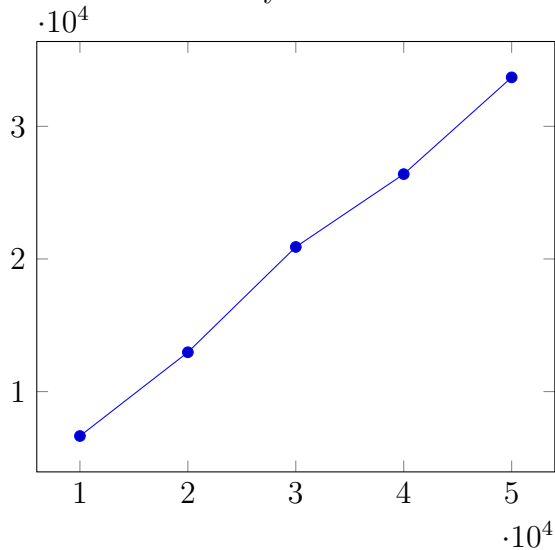
## Дневник отладки

1. Возникли сложности с работой чекера и `Makefile`, было принято решение переписать программу в один файл, чтобы не тратить попытки напрасно.
2. Превышено максимальное время выполнения на тесте 13, было принято решение отключить синхронизацию ввода-вывода с библиотекой `stdio`

3. Превышено максимальное количество памяти, было принято решение в массиве `ans` хранить не пару «ключ - значение», а индекс элемента в исходном массиве, таким образом уменьшили занимаемый массивом `ans` объём памяти.

## Тест производительности

Ниже приведён график зависимости времени выполнения программы (по оси  $Y$  в миллисекундах) от количества пар «ключ - значение» (по оси  $X$ , в тысячах), как мы видим, сложность соответствует линейной.



## Недочёты

Единственным недочётом является то, что вся программа сделана в одном файле, хотя можно было бы, например, реализацию класса `TPair` вынести в отдельный заголовочный файл. Как отмечалось ранее, на это пришлось пойти из-за проблем с чекером. Тем не менее, программа не выглядит из-за этого громоздко, так как её размер не превысил 80 строк.

## Выводы

При выполнении лабораторной работы была написана программа сортировки за линейное время. Она может быть полезна в ситуации, когда во входящей последовательности много повторяющихся ключей, по которым сортируются элементы. Также важно помнить, что сортировка подсчётом хоть и работает за линейное время, но требует дополнительную память.