

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

Выполнил студент группы М8О-212Б-22 МАИ *Юрков Евгений*.

## Условие

### Вариант: 7-1

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 определяется типом ключа (и соответствующим ему методом сортировки) и типом значения: Поразрядная сортировка.

Тип ключа: автомобильные номера в формате А 999 ВС (используются буквы латинского алфавита).

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 поразрядной сортировки строки происходит сортировка подсчетом массива пар ключ-значение. По очереди в функцию сортировки подсчетом передается сортируемый индекс от большего к меньшему (от 7 до 0, так как строки фиксированной длины). Для сортировки подсчетом создается массив **counter** содержащий количество повторений каждого символа на позиции **i**. Далее происходит расстановка ключей вместе с соответствующими им значениями в отсортированном порядке в новом массиве **result**. На последнем шаге происходит перемещение информации в исходный массив.

## Описание программы

Данные хранятся в структуре `std::pair<std::string, std::string>`. Программа состоит из следующих функций:

1. `void __count_sort(std::vector<std::pair<std::string, std::string> >& arr, uint index)` - функция осуществляющая сортировку подсчетом индекса `index` в массиве `arr`.
2. `void radix_sort(std::vector<std::pair<std::string, std::string> >& arr)` - функция поразрядной сортировки для строки, сортирует значения строк в массиве передавая индексы в функцию сортировки подсчетом.
3. `std::pair<std::string, std::string> parse(const std::string& str)` - преобразует введенную строку в структуру ключ-значение.
4. `int main()`.

## Дневник отладки

- Сначала не были учтены пустые введенные строки и ключи без значений.
- Для преодоления ML во всех присваиваниях строк и массивов в функции сортировки подсчетом был использован `std::move( )`.
- Сначала использовалась поразрядная сортировка для чисел, но для преодоления TL функция сортировки была переписана специально для сортировки строк длины 8.
- Для сокращения времени перебора массива `counter` его размер был сокращен с 256 до ('Z' - '0' + 1) (теперь в нем хранятся счетчики только тех символов, которые могут использоваться).

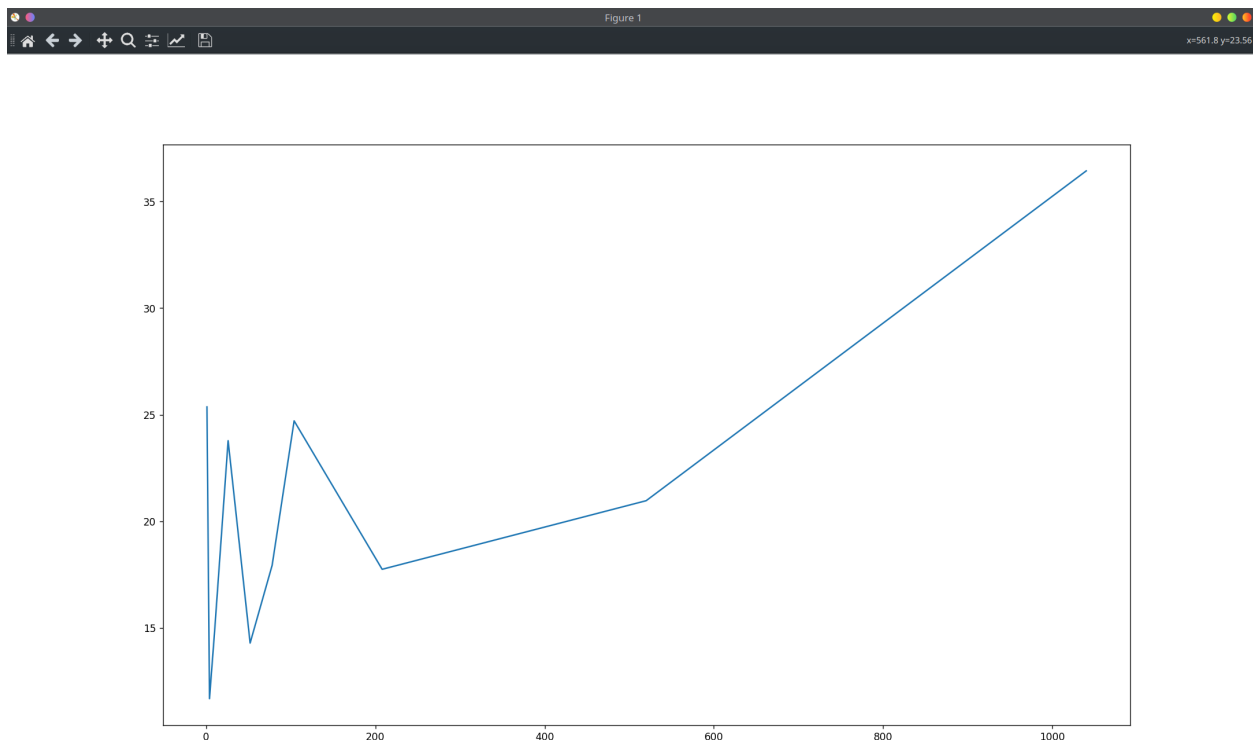


Рис. 1: График зависимости времени работы программы от количества введенных данных

### Тест производительности

Сложность написанного алгоритма  $O(n)$ . Для построения графика (Рис. 1) использовались тесты с 1 - 1000 строками данных. Из графика видно, что количество входных данных слабо влияет на время работы программы.

## Выводы

Поразрядная сортировка важна для эффективной обработки больших объемов данных, когда необходимо быстро выполнить сортировку. Она позволяет обеспечить линейную сложность алгоритма относительно количества элементов в массиве, что делает ее особенно полезной для больших и расширяющихся наборов данных. Однако этот метод сортировки неприменим для сложных структур данных, в этом он уступает сортировкам за  $O(n \log n)$ , для которых достаточно определить операцию сравнения элементов.