

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

Выполнил студент группы М8О-215Б-23 МАИ *Дехтеренко Дмитрий*.

## Условие

Поразрядная сортировка.

Тип ключа: телефонные номера, с кодами стран и городов в формате +<код страны> <код города> телефон.

Тип значения: строки переменной длины (до 2048 символов).

## Метод решения

Для решения задачи используется поразрядная сортировка. Вначале разрядами считаются код страны(старший), код города и телефон(младший). В соответствии с алгоритмом, для линейной сложности мы должны отсортировать массив устойчивой сортировкой с той же линейной сложностью каждый разряд, начиная с младшего. Этой сортировкой будет та же поразрядная сортировка, содержащую внутри сортировку с линейным временем работы. Далее сортируем уже числа в каждом из трёх полей. Разрядами будут байты числа. Сортировать их для линейной сложности будем сортировкой подсчётом. Получается двухэтапная поразрядная сортировка.

## Описание программы

Типы данных:

- Так как количество данных было заранее неизвестно, был написан свой шаблонный вектор **TVector<T>** для использования со структурами **TPhoneNumber** и символами.
- В основной структуре **TPhoneNumber** присутствуют числовые поля кодов, строка-телефон для вывода и динамическая строка-значение. Для сортировки были выбраны числа, потому что с ними удобнее писать функцию сортировки, которая будет единообразно сортировать по тому или иному полю. Недостатком такого решения является возможное наличие в кодах ведущих нулей, информация о которых пропадает при сохранении значений в целочисленном формате, и которую нужно как-то отдельно зафиксировать, что приводит к перерасходу памяти.

Функции:

- В функции **RadixSort** в количестве раз, равном количеству разрядов, происходит сортировка подсчётом переданного массива. После сортировки каждый раз в массиве числа сдвигаются на заданное число бит вправо.

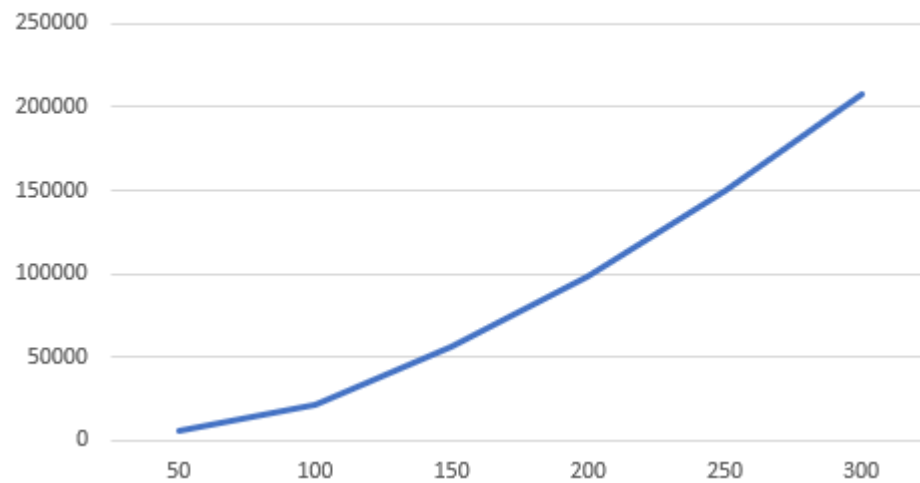
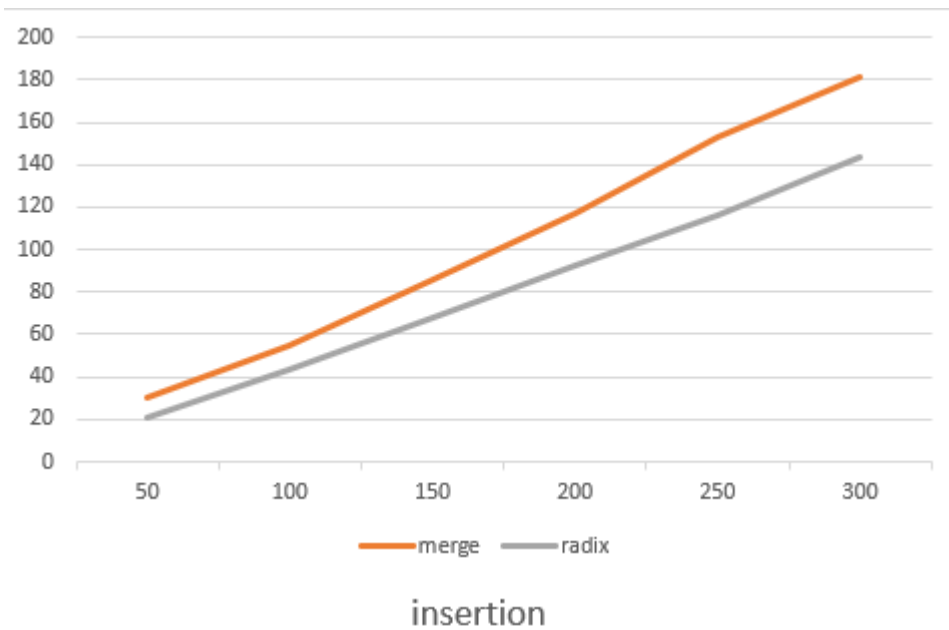
- В функции **CountingSort** формируется массив из младшего бита чисел, и с ним производится сортировка подсчётом. Так же, так как сортировка подсчётом требует дополнительной памяти, создаётся не дополнительный массив структур, занимающий огромный объём памяти, а массив позиций.

## Дневник отладки

Дата	Событие	Действие по исправлению
12.04.2025	Не подумал, что у кодов могут быть ведущие нули	Добавил в структуру строку для вывода с ведущими нулями
12.04.2025	Получил Memory Limit	Стал работать в функциях сортировки с массивами позиций, а не с массивами структур; строка-значение в структуре теперь динамический массив
13.04.2025	Неправильно сохранял строку-значение в структуру	Стал передавать корректную длину строки в метод SetData
13.04.2025	Снова получил Memory Limit	Создавал массив векторов строк-значений, а не один вектор в структуре; поправил

## Тест производительности

Для сравнения необходимо было выбрать так же устойчивые сортировки, потому что на первом этапе при их использовании в данной задаче необходима всё так же поразрядная сортировка по 3 полям, просто внутри неё уже будет сортировка с нелинейной сложностью. Алгоритм сравнивался с сортировкой слиянием и сортировкой вставками. Сортировки тестировались с наборами случайных строк, подходящими под условие, размерами в 50, 100, 150, 200, 250 и 300 тыс. строк. Результаты тестов приведены на графиках (на оси времени миллисекунды):



Поразрядная сортировка показала линейную сложность, но не показала значительного преимущества над сортировкой слиянием несмотря на нелинейность последней, сильный разрыв будет виден только на огромных массивах данных. Сортировка вставками же находится далеко позади с квадратичной сложностью.

## Выводы

Алгоритм подходит для случаев, когда нужно отсортировать целые числа, и допускается использование дополнительной памяти.

Было сложно и интересно думать над оптимизацией алгоритма, чтобы он проходил тесты.