

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

Выполнила студентка группы 08-307 МАИ Усачева Елизавета.

Условие

Необходимо написать программу, осуществляющую ввод пар ключ-значение, упорядочить их по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывести отсортированную последовательность.

1. На каждой непустой строке входного файла располагается пара ключ-значение, затем следует знак табуляции и указано соответствующее значение. Выходные данные состоят из тех же строк, что и входные, за исключением пустых и порядка следования.
2. Вариант задания: 7-3.

Ключ Автомобильные номера в формате A 999 BC (используются буквы латинского алфавита).

Значение Числа от 0 до $2^{64} - 1$.

Метод решения

- С помощью цикла `while` (до наступления «End of File» или ввода команды окончания считывания) через стандартный ввод `stdin` осуществляется «бесконечный» ввод данных. Бесконечный в том смысле, что заранее неизвестно, сколько будет введено данных. В этом удобство данного решения.
- В основе метода поразрядной сортировки я использую метод сортировки подсчетом, который реализован в функции *RadixSort* моей программы.
- Хранение пар ключ-значение реализовано с помощью массива структур, которые содержат два поля: ключ и значение. Поле *ключ* - это массив типа *char* размером [7], поле *значение* - переменная типа *unsigned long long*.
- Я работаю со строками от последнего символа к первому. В каждой итерации создается вспомогательный массив от 0 до 91, соответствующий десятичным кодам ASCII. Этот массив сначала заполняется нулями. Затем, происходит проход по исходному массиву и считается количество его *i*-ых элементов, которое записывается в вспомогательный массив. Следующим шагом будет подсчет элементов, не превосходящих *j*-го в вспомогательном массиве. Затем, на основе предыдущего

шага, каждый элемент исходного массива помещается на нужное место во временном массиве. И наконец, используя функцию *memcpy* я присваиваю исходному массиву адрес памяти, который содержит ссылку на первый элемент нового отсортированного массива.

Описание программы

- **radix_sort.c**

Основной файл, написан на языке C. Содержит в себе функцию *main*, функцию *RadixSort*, описание используемых структуры и констант.

Дневник отладки

- Самой первой проблемой было корректное считывание специфицированных данных. Я пользовалась функцией *scanf* с различными спецификаторами. Однако, последний символ строки всегда забивался мусором. Решением этой проблемы было принудительное установление в конец строки символа конца строки. В дальнейшем я решила использовать функцию *getchar*, т.к. она оказалась быстрее.
- Все последующие проблемы были связаны с неверной работой с динамической памятью. Происходила утечка памяти.

Выводы

Данный алгоритм поразрядной сортировки основывается на методе сортировки подсчетом, который подразумевает создание вспомогательных массивов. Поэтому областью применения являются данные, которые имеют определенный и конечный диапазон значений, например целые числа или, как в моем случае, символы, имеющие соответствующие им коды ASCII. В отличие от многих других алгоритмов сортировки, поразрядная сортировка не использует операцию сравнения элементов, что выгодно его отличает. Однако, при сортировке, например, длинных строк с большим диапазоном значений, в каждой итерации будет создаваться вспомогательный массив, что будет отрицательно влиять на потребление памяти.