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

Выполнил студент группы 08-208 МАИ *Армишев Кирилл.*

Условие

Кратко описывается задача:

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

   (a) Поразрядная сортировка.

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

   (c) 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

Алгоритм поразрядной сортировки достаточно прост. Достаточно пройтись по всем разрядам справа налево (Кормен "Алгоритмы. Построение и анализ"глава 8.3 "Поразрядная сортировка") и применить устойчивую сортировку. Лично я использовал сортировку подсчетом, так в данном случае k«n(где k - максимальное значение сравниваемых значений(от 0 до 9), а n - кол-во сравниваемых значений).

Описание программы

Весь код содержится в файле **main.c**: структура данных **Record**, в которой хранится ключ(char phone[21]), указатель на значение(char* value), и phone_long(в нем хранится номер телефона без + и -).

Дневник отладки

С самого начала я решил реализовать обычную поразрядную сортировку с помощью сортировки подсчетом каждого разряда и потратил на это около 40 тестов, не замечая данной фатальной ошибки и исправляя в программе совсем не то. Затем я перечитал

конспект лекции Никиты Константиновича, понял, что намного эффективней представлять наши значения в виде последовательности битов. Я реализовал побитовую поразрядную сортировку, улучшил расход памяти и программа прошла все тесты.

Тест производительности

N	Time (microseconds)
100	341
1000	2898
10000	24092
20000	49417
50000	104547
100000	186924

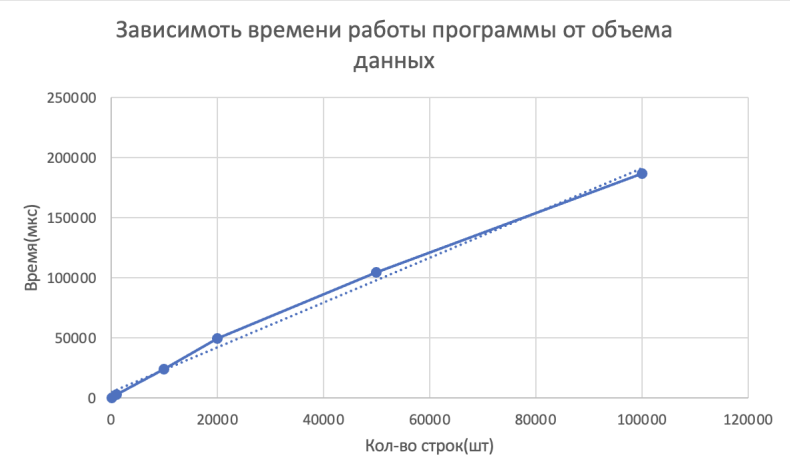

Рис. 1: Зависимоть времени работы программы от объема данных

Выводы

Известно, что сортировка подсчетом эффективна лишь тогда, когда k<=n. Поэтому использовать ее для поразрядной сортировки возможно. В данном варианте обычная поразрядная сортировка не эффективна. Следует применять побитовую поразрядную сортировку, представлять каждый ключ в виде последовательности битов и уже к данной последовательности применять сортировку подсчетом. Это намного эффективней, так как битовые маски быстрее считаются. Сложность такой сортировки будет O(d(n+2^R-1)). Как видно из теста производительности, сложность сортировки практически O(n).