

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

Выполнил студент группы М08-312Б МАИ *Лобанов Олег*.

Условие

Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.

Вариант сортировки: Поразрядная сортировка

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

Вариант 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

Числа сортируются сортировкой подсчетом по каждому разряду. В нашем случае сортировка сначала сортирует младшие разряды, затем старшие.

Для чисел удобнее использовать разряды не в десятичном представлении, а несколько битов двоичного представления. Так как для получения цифры числа в двоичном представлении используются легкие операции логического сдвига.

Описание программы

Согласно Кормену для n b -битовых чисел и цифр из r битов алгоритм поразрядной сортировки выполнит работу за $O(\frac{b}{r}(n + 2^r))$. Выбрать r нужно в зависимости от $\log_2 n$. Если $b < \log_2 n$, то $r = b$, иначе $r = \log_2 n$.

Сперва создается временный массив B равный исходному, затем массив чисел C размером 2^r в котором будет проходить сортировка подсчетом. Число разбивается на непересекающиеся части по r бит и сортируется подсчетом по ним, начиная с младшего разряда.

Совершается проход по исходному массиву, для каждой цифры значение элемента C на позиции со значением этой цифры увеличивается на 1. Затем элементы массива C последовательно суммируются, таким образом в массиве C значение по индексу цифры равно количеству строго меньших чем этот индекс элементов. Идем по изначальному массиву назад (для сохранения устойчивости): для j элемента этого массива в $C[B[j] - 1]$ содержится корректный индекс этого элемента в массиве. Уменьшаем значение $C[B[j]]$ на 1 чтобы следующий элемент с таким же значением не записался поверх этого.

Копируем в вспомогательный массив отсортированный по текущему разряду массив и сортируем по следующему.

Дневник отладки

На первой попытке произошла ошибка компиляции, где забыл прописать `std`. Со второй попытки код "окнулся". Во время локального тестирования погуглил про форматирование вывода.

Тест производительности

Сравниваю производительность поразрядной сортировки со стандартной библиотечной *std::stable_sort*.

Для 1к элемметов:

Radix - 0.470ms

Stable_sort - 0.400ms

Для 1кк элементов:

Radix - 520.879ms

Stable_sort - 640.435ms

Для 15кк элкметов:

Radix - 8609.285ms

Stable_sort - 11750.267ms

Выводы

Выполнив первую лабораторную работу, я изучил сортировки за линейное время, осознал их преимущества и недостатки. Опробовал на практике ввод и вывод данных на C++, методы `std::istream`.