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

Выполнил студент группы 08-207 МАИ *Хренов Геннадий.*

Условие

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: Поразрядная сортировка. Тип ключа: телефонные номера, с кодами стран и городов в формате +<код страны> <код города> телефон. 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 решения задачи был создан класс вектор, где хранились входные данные в структуре, состоящей из телефона, значения и числа-телефона(телефон без знаков разделения разрядов). После считывания ключа-значения заполняется число-телефон, затем эти числа сортируются методом поразрядной сортировки, которая составляет N сортировок подсчётом от младшего разряда к старшему, где N - максимальное количество разрядов в числе. Сортировка подсчётом требует создания дополнительного массива из 10 элементов и вектора с длиной входного, куда запишется результат сортировки.
Брал информацию из:
radix sort [Электронный ресурс] URL: https://www.geeksforgeeks.org/radix-sort/
Кормен - Алгоритмы: построение и анализ.

Описание программы

Программа состоит из одного файла 2ex.cpp. Основные классы и функции:
class TVector: описание класса вектора с его функциями
TVector() - конструктор
int Size() - возвращает размер вектора
friend void swap(TVector lhs, TVector rhs) - меняет местами содержимое двух векторов
valueType operator[](int index) - пергрузка оператора, возвращает значение для заданного индекса
TVector operator=(TVector other) - перегрузка присваивания для векторов
void PushBack(const valueType value) - добавляет элемент, а в случае переполнения вектора создает копию в 2 раза большего размера и перемещает элементы туда
struct TKv - структура, состаящая из телефона(массив чаров), значения(массив чаров)

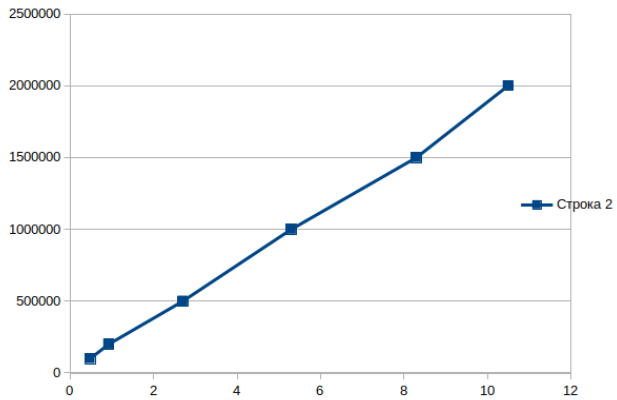
и числа для сортировки(unsigned long long)
void CountingSort(TVector<TKv> elems, unsigned long long q) - функция, выполняющая сортировку подсчётом
void RadixSort(TVector<TKv> elems, unsigned long long max1) - функция, выполняющая цифровую сортировку
long long Maxi(TVector<TKv> qu) - возвращает максимальное число

Дневник отладки

Сначала написал программу для сортировки строк. Работает правильно, но алгоритм не эффективный:
1)ошибка компиляции - переделал сортировки так, что аргументов на вход стало по 2.
2-3) превышен лимит времени - пробовал рассинхронизацию и scanf(неудачно). Переписал программу для сортировки чисел.
4) неверный ответ - переполнилась интовая переменная цикла, после изменения её на llu все тесты прошли(9 + 11 с кодстайлом)
5-8) ещё один алгоритм сортировки строк, который полагает максимально длинный телефон из 17 символов и приводит к нему все телефоны приписывая нули вначале. Ошибка выполнения - заменял размер массива для входной строки.

Тест производительности

время	элементы
0,475	100000
0,92	200000
2,7	500000
5,3	1000000
8,3	1500000
10,5	2000000



По графику видно, что сортировка действительно совершается примерно за линейное время.

Недочёты

Программа работает правильно, однако использует сортировку чисел, причём возникает необходимость использовать оператор остатка от деления, а это операция является сложной для компьютера. Хочу заметить, что это программа немного проигрывает по времени программе сортирующей строки(на 3 секунды на 2 млн. элемнтах).

Выводы

Поразрядная сортировка часто используется для упорядочения данных с несколькими полями. Например, телефонные номера, даты и тд.. Radix sort может использовать не только сортировку подсчётом, но и другие, однако важно, чтобы эти алгоритмы были устойчивыми. Мне стоило более детально разобраться с условием задания и сразу выбирать алгоритм наиболее подходящий для успешного выполнения работы. Это бы уменьшило время выполнения лабораторной работы в целом. Поразрядная сортировка проста в написании, однако, если она опирается на сортировку подсчётом, потребуется дополнительный массив, поэтому в случае, когда требуется экономить память, это не лучший выбор.