

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Институт № 8 информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работ №2 по курсу
«Дискретный анализ»

Поразрядная сортировка

Студент: Пермяков Никита Александрович
Группа: М80 – 208Б-19
Вариант: 6-1
Преподаватель: Кухтичев Антон Алексеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2020

Содержание

1. Постановка задачи
2. Метод и алгоритм решения
3. Сведения о программе
4. Основные файлы программы
5. Демонстрация работы программы
6. Вывод

Постановка задачи.

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: 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 и алгоритм 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.

Общие сведения о программе

Программа компилируется из одного файла: main.cpp. Не используются заголовочные файлы.

`class TVector` – класс для указателя на структуру ключа и текущего размера с методами динамического изменения размера выделенной памяти, и перегруженным оператором `[]` для доступа к структуре ключа.

`struct TValue` – структура для значения тела строки

`struct TKey` – структура для значения ключа строки

`struct TKey_value` – структура ключа с полями для кодов телефона и индексом

`void GetInputIndex(TKey_value* value, int i)` – возвращает индекс текущей структуры (может не совпадать с номером, по которому вызывается структура)

`char StorageIndex(TVector<TKey_value> output, int j, TKey *keys, int level, int rank)` – возвращает ключ с учетом смещения по уровню (части номера телефона) и номера элемента в строке.

`TVector<TKey_value> Counting(int rank, TVector<TKey_value> output, TVector<TKey_value> tmp, int count, TKey *keys, int level)` – функция, выполняющая сортировку подсчетом

`int LevelSize(TVector<TKey_value> output, int j, int level, int rank)` – Возвращает текущий размер (не обработанной) части номера телефона.

`int Sum_size(int count, TVector<TKey_value> output, int i)` – суммы частей ключа по уровню начиная с кода страны

Дневник отладки.

Сначала написал программу для сортировки строк. Работает правильно, но алгоритм не эффективный:

1-7) ошибка компиляции - переделал сортировки так, что аргументов на вход стало по 2.

8-13) превышен лимит времени - пробовал рассинхронизацию и scanf (неудачно). Переписал программу для сортировки чисел.

14) неверный ответ - переполнилась интовая переменная цикла, после изменения её на llu все тесты прошли(9 + 11 с кодстайлом)

15) 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.

Далее рефакторинг и кодстайл.

Недочеты.

Программа работает правильно, однако использует сортировку чисел, причём не возникает необходимость использовать оператор остатка от деления, в то время как это операция является сложной для компьютера.

Вывод

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.