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

Выполнил студент МАИ группы М8О-201Б *Ефимов Александр.*

Постановка задачи

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

Вариант 5-1

- Тип сортировки: поразрядная сортировка.
- Тип ключа: MD5-суммы (32-разрядные шестнадцатиричные числа).
- 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

Сама программа должна состоять из трех частей:

1. Реализации вектора для хранения неопределенного количества данных
2. Сортировки соответствующего вектора
3. Ввода в вектор и вывода из него

Так как работа происходит с шестнадцатиричным числами, то, для удобства, они вводятся как массивы символов, а позже переводятся в соответствующие значения в десятичной системе при помощи оперций над символами.
Поразраядная сортировка происходит за счет посчета количества цифр в одном и том же разряде каждого ключа за один проход и записывания их количеств в массив чисел размером с основанием системы исчисления (например, если в втором разряде первых 5 чисел два нуля и три единицы, то в нулевую позицию массива записывается 2, во первую - 3).

После этого из нулевой позиции вычитается единица, а каждое следующее число заменяется на его сумму с предыдущим ему числом.
В итоге у нас получается массив, указывающий на позиции каждого ключа в отсортированном векторе

Для расстановки ключей в отсортированный вектор достаточно начать обход текущего вектора с конца до начала включительно, а ключи записывать в позицию, равную числу в ячейке массива под номером, равного цифре разряда. После записи из числа вычисть единицу.

Описание программы

1. **Vector.h**

   Определяет шаблонный вектор класс

   - Определены конструкторы для класса, а также деструктор;
   - Swap функция для замены местами двух векторов;
   - Перегружаны операторы приравнивания для LValue и RValue, а также оператор индексации;
   - Методы, возвращающие указатели на начало и конец;
   - Методы, определяющие размер вектора и его пустоту (empty).

2. **Sort.cpp**

   - Три константы показывающие на максимальные размеры строки, ключа, а также основание системы исчисления соответственно;
   - Тип Slot, который содержит в себе ключ и соответствующую ему строку;
   - Сама функция сортировки;
   - *int main()*.

Дневник отладки

Номер	Ошибка	Обнаруженная причина	Исправление
1	Ошибка выполнения	*return 1* при пустом векторе	Заменить на *return 0*
2	Неправильный ответ	Вывод текста при пустом векторе	Убрать вывод текста
3	Превышено реальное время работы	Неверно написан шаг алгоритма сортировки	Исправление шага
4	Ошибка выполнения	Переполнение массива посчета цифр	Изменить тип с *short* на *unsigned int*

Тест производительности

На рисунках 1-3 показаны разные входящие данные и времена их сортировки. Игнорируя артефакты, вызванные особенностями работы системы, можно прийти к выводу, что данные сортируются со сравнительно одинаковой скоростью согласно сложности $O(d * (n + k))$, где d – число цифр, а k – основание системы исчисления.
Так как d и k константы, то сложность – линейная.

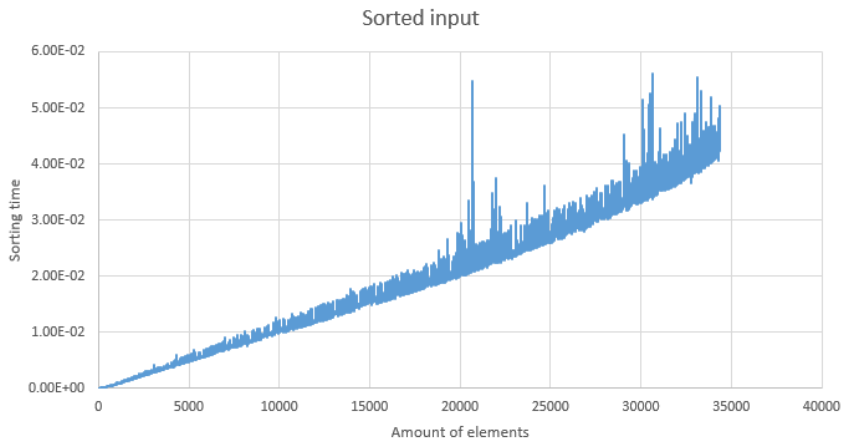

Рис. 1: Отсортированные входящие данные

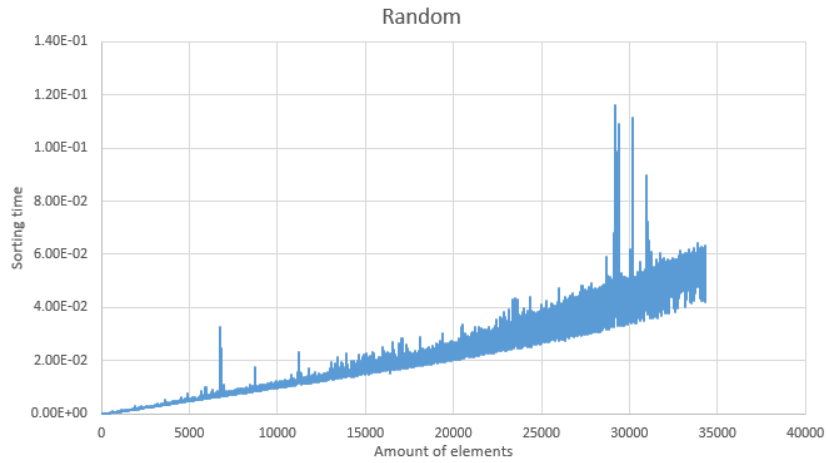

Рис. 2: Случайные входящие данные

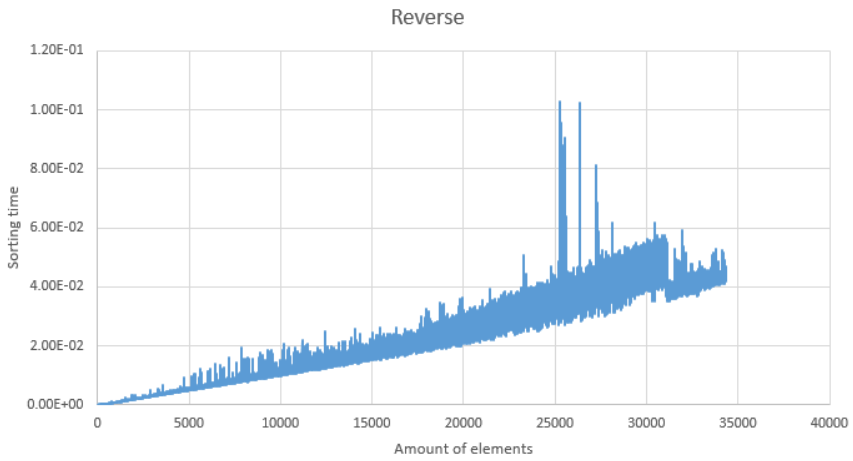

Рис. 3: Обратно отсортированные входящие данные

Недочёты

В связи с особенностью сортировки, а именно массива подсчета цифр, без усложнения алгоритма сортировки и/или ввода данных, количество вводимых строк ограничено размером типа этого массива (в частности, для массива типа *unsigned long long* количество данных ограничено до 2^{64} ключей и строк).

Выводы

Поразрядная сортировка может быть использована на месте сортировки подсчетом там, где диапазон вводимых данных может быть слишком большим для сортировки подсчетом. Также, поразрядная сортировка часто используется при сортировки карт и может быть использована при сортировки слов (при этом начиная с первого разряда, а не с последнего).