

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

Выполнил студент группы М80-208Б-18 МАИ *Алексеева Мария*.

Условие

1. Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа сортировкой подсчетом и вывод отсортированной последовательности.

2. Вариант 2-2.

Сортировка подсчетом.

Тип ключа: почтовые индексы.

Тип значения: строка до 2048 символов.

Метод решения

1. Данные на вход программе подаются через стандартный поток ввода,

```
std::istream &operator>>(std::istream &in, TValue &m) {  
    in >> m.key >> m.value;  
    return in;  
}
```

2. Так как количество данных заранее не известно, необходимо будет реализовать вектор (по условию задания стандартный вектор из библиотеки STL использовать запрещается).
3. Ключи будут представляться в виде беззнакового числа и в процессе ввода будет видоизменять число, заполняя его нулями при необходимости. out « std::setw(6) » « std::setfill('0') » « m.key;»
4. Сортировка подсчётом принимает тот же вектор входных данных и максимальное значение ключа, а также создаёт 2 новых массива: **b**, в который будет записана отсортированная последовательность, и **c**, в котором подсчитывается число повторений каждого возможного символа в сортируемом разряде ключа.

Описание программы

На каждой непустой строке входного файла располагается пара «ключ-значение», поэтому создадим новую структуру **TKeyValue**, в которой будем хранить ключ и значение.

main.cpp	
Тип данных	Значение
struct TKeyValuePair	Структура для хранения пары "ключ-значение"
class TVector	Вектор для хранения структур TData
Функция	Значение
TVector()	Конструктор класса TVector
TVector(size _t)	Конструктор класса TVector
~ TVector()	Деструктор класса TVector
void PushBack(TData)	Добавить элемент в конец вектора
void CountingSort(size _t)	Функция сортировки подсчётом
int main()	Главная функция, в которой происходит чтение данных, вызов функции сортировки и вывод.

Дневник отладки

При создании этой таблицы была использована история посылок.

время	ошибка	причина
2019/11/30 15:15:00	Превышено время	В главной функции создавался новый результирующий вектор, который копировал отсортированный, что замедляло время работы программы.
2019/11/30 15:16:49	Превышено реальное время работы	Проблема в неоптимизированном выводе. Использованные методы оптимизации: 1) Опция <code>std::ios_base::sync_with_stdio(false)</code> , + +, ...2) <code>std::cin.tie(nullptr).cout</code> ..
2019/12/05 12:04:49	Превышено реальное время работы	Изменено редактирование ключа с помощью функций <code>std::setw(6)</code> и <code>std::setfill('0')</code> . До этого использовался массив из 6-ти элементов.
2019/12/05 12:16:07	Работает	Далее убирались комментарии и лишние функции, используемые только для отладки программы, вводились незначительные косметические изменения.

Тест производительности

Для генерации тестов использовалась следующая программа:

```
#include <iostream>
#include <fstream>
#include <ctime>
#include <cmath>
```

```

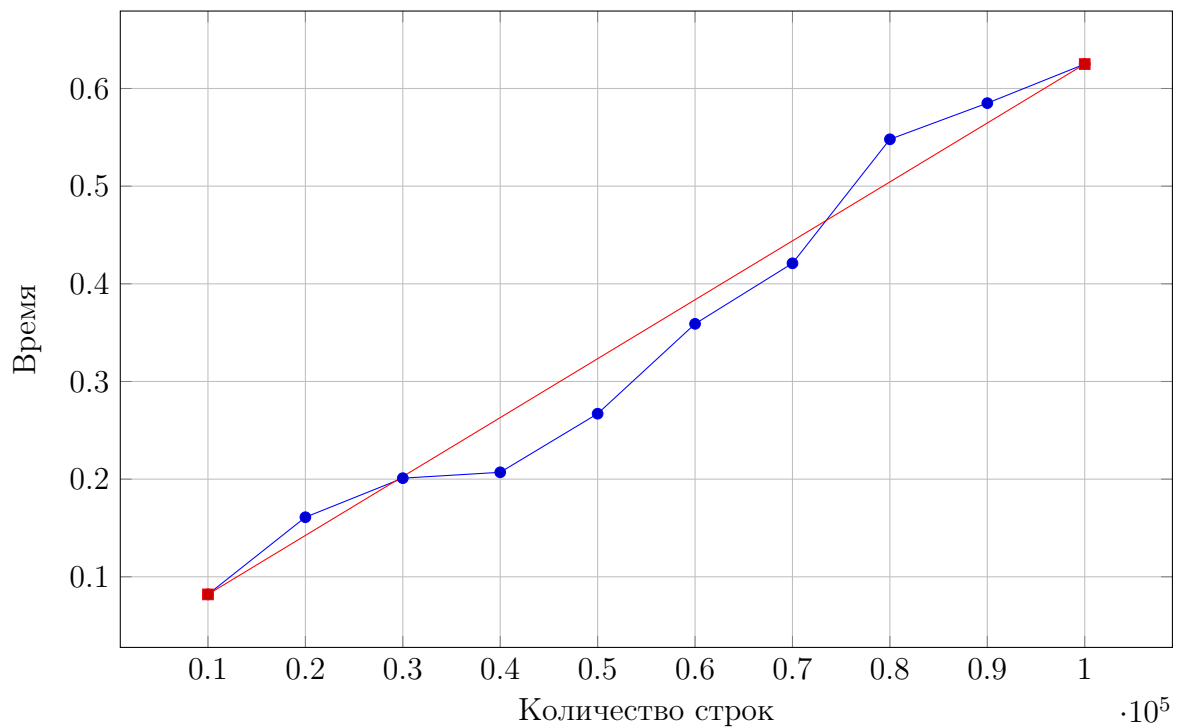
#include <cstdlib>

int main(int argc, char *argv[]) {
    std::ofstream file(argv[1]);
    srand(time(0));
    size_t size = atoi(argv[2]);
    char key[6];

    for (int i = 0; i < size; i++) {
        for (int j = 0; j < 6; j++) {
            key[j] = (char) (rand() % 10 + 48);
        }

        int length = rand() % 2050;
        char data[length];
        for (int k = 0; k < length-1; k++) {
            data[k] = (char) (rand() % 26 + 97);
        }
        data[length-1] = '\\0';
        file << key << '\\t' << data << '\\n';
    }
    return 0;
}

```



Выводы

Данную программу можно использовать для сортировки файлов по ключу, который представляет собой постовый индекс.

В процессе работы я получила ценный опыт отладки программы и её оптимизации при помощи утилиты `valgrind`. Также разобралась в работе сортировки подсчетом, научилась различать устойчивые сортировки от неустойчивых.