

Московский Авиационный Институт  
(Национальный Исследовательский Университет)

**Факультет «Компьютерные науки и прикладная математика»**

**Кафедра вычислительной математики и программирования**

**Лабораторная работа №1 по курсу  
«Дискретный анализ»**

**Тема работы  
“Сортировки за линейное время”**

Студент : А.А. Дудовцев

Группа : М8О-208Б-22

Оценка : \_\_\_\_\_

Дата : \_\_\_\_\_

Подпись : \_\_\_\_\_

Москва, 2024

## 1. Постановка задачи

1. В лабораторной работе было необходимо реализовать сортировку за линейное время, вводимых пар (ключ – значение).
2. Вариант задания (А. 1-1): моим вариантов предусматривается сортировка подсчетом, ключом выступает целое число от 0 до 65535, а значением строка длиной до 64 символов.

## 2. Метод решения:

Алгоритм решения: сначала мы заводим структуру данных “pair”, где будет два поля (ключ, значение). Далее считываем до EOF все пары из входного потока данных в массив пар (pairs), динамически его расширяя (realloc). После чего вызываем функцию сортировки подсчетом, которая реализована следующим образом: создаем новый массив длиной максимального ключа и считаем сколько раз встречается каждый ключ в считанных парах. После чего преобразуем его в массив префиксных сумм, тогда в этом массиве под индексом ключа будет находиться место, на которое мы должны будем поставить элемент после сортировки. Далее создаем финальный массив, в который добавляем наши пары, в соответствии с индексом в массиве префиксных сумм, а затем переобозначаем область памяти в изначальном массиве, на область памяти отсортированного массива (с помощью memmove). Далее выводим результат в консоль.

## 3. Описание программы:

Архитектура программы достаточно проста - всего один исполняемый файл. Используется всего одна функция типа void – CountingSort(), которая принимает на вход указатель на неотсортированный массив пар и количество элементов в массиве. Помимо этого реализована структура “pair”, два поля которой являются значением и ключом.

## 4. Дневник отладки:

Сначала задача не зашла, потому что в выводе неправильно учел табуляцию: выводил ответы через пробел а нужно было через таб. После исправления этого момента, программа успешно зашла.

## 5. Исходный код

```
#include <iostream>
#include <string.h>

const int MAX_KEY_VALUE = 65535;

struct Pair {
```

```

    int key;
    char value[65];
};

void scan_array(Pair **array, int array_size, int *pair_counter) {
    while (std::cin >> (*array)[*pair_counter].key >>
(*array)[*pair_counter].value) {
        ++(*pair_counter);
        if ((*pair_counter) == array_size) {
            array_size *= 2;
            (*array) = (Pair *)realloc((*array), array_size * sizeof(Pair));
        }
    }
}

void print_array(Pair *array, int pair_counter) {
    for (int i = 0; i < pair_counter; ++i) {
        std::cout << array[i].key << '\t' << array[i].value << '\n';
    }
}

void count_sort(Pair *array, Pair *new_array, int pair_counter) {
    int frequency[MAX_KEY_VALUE + 1];
    memset(frequency, 0, (MAX_KEY_VALUE + 1)*sizeof(int));

    for (int i = 0; i < pair_counter; ++i) {
        ++frequency[array[i].key];
    }

    int total_counter = 0, old_counter;
    for (int i = 0; i <= MAX_KEY_VALUE; ++i) {
        old_counter = frequency[i];
        frequency[i] = total_counter;
        total_counter += old_counter;
    }

    for (int i = 0; i < pair_counter; ++i) {
        new_array[frequency[array[i].key]] = array[i];
        ++frequency[array[i].key];
    }
}

```

```

    }
}

int main() {
    int pair_counter = 0;

    Pair *array = (Pair *)malloc(sizeof(Pair));
    scan_array(&array, 1, &pair_counter);
    Pair *new_array = (Pair *)malloc(pair_counter * sizeof(Pair));
    count_sort(array, new_array, pair_counter);
    print_array(new_array, pair_counter);

    free(array);
    free(new_array);

    return 0;
}

```

## 5. Тест производительности:

На тестах программа удовлетворяет заявленной сложности.

## 6. Выводы:

Данный алгоритм применим в различных задачах, где необходима сортировка по одному из значений пары данных, например, найти имена людей, которые старше определенного числа лет. Путем реализованной выше сортировки, мы с легкостью найдем таких людей, ну и т.д. В принципе, особых сложностей не возникло при выполнении лабораторной работы.