Отчет по лабораторной работе № 2 по курсу “Фундаментальная информатика”

Студент группы М80-101Б-21 Ершова Станислава Григорьевича, № по списку 8

Контакты e-mail, telegram: stas.ershov57@gmail.com , @stas_orel

Работа выполнена: «2» декобря 2020г.

Преподаватель: каф. 806 Севастьянов Виктор Сергеевич

Отчет сдан «2» декобря 2021 г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема: Сортировка и поиск**

2. **Цель работы: Научиться использовать сортировки**

3. **Задание** (*вариант №* **7**)**:** Сортировка Шелла

4. **Оборудование** (студента):
   Процессор *Intel Pentium N4200 1.1 ГГц* с ОП *8* Гб, SSD *128* Гб. Монитор *1920x1080*

5. **Программное обеспечение (**студента**):**
   Операционная система семейства: *linux*, наименование: *ubuntu*, версия *20.04.3 LTS*
   интерпретатор команд: *bash* версия *5.0.17*
   Система программирования -- версия --**,** редактор текстов *nano* версия *4.8*
   Утилиты операционной системы --
   Прикладные системы и программы --
   Местонахождение и имена файлов программ и данных на домашнем компьютере --

6. Идея, метод, алгоритм:

7. Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <time.h>


typedef struct data_st
{
   int key;
   char value[256];
} data_st;

void swap(data_st *a, data_st*b) {
   data_st temp = *a;
   *a = *b;
   *b = temp;
}

//O(n^(3/2))
void shell_sort(data_st data[256], int input_size, int reversed) { // 1: min -> ... -> max | -1: max -> ... -> min
   int step = input_size / 2;
   while (step >= 0) {
      for (int i = 0; i + step < input_size; ++i) {
         int ind = i + step;
         while(ind >= i + step && data[ind].key * reversed < data[ind - step].key * reversed) {
            swap(&data[ind - step], &data[ind]);
            ind -= step;
         }
```

```
        }
        step--;
    }
}


void print_struct(data_st data[256], int input_size) {
    for (int i = 0; i < input_size; ++i)
        printf("%d %s", data[i].key, data[i].value);
}


void vector_shuffle(data_st data[256], int input_size) {
    long long step = time(NULL) % 4 + 1;
    for (long long i = step; i < input_size; ++i) {
        long long ind = i;
        while (ind >= step) {
            swap(&data[ind - step], &data[ind]);
            ind -= step;
        }
    }

    step = (time(NULL) / 83 * 56 + 215) % 2 + 1;
    for (long long i = step; i < input_size; ++i) {
        long long ind = i;
        while (ind >= step) {
            swap(&data[ind - step], &data[ind]);
            ind -= step;
        }
    }

    step = (time(NULL) / 15 * 9 + 20) % 3 + 1;
    for (long long i = step; i < input_size; ++i) {
        long long ind = i;
        while (ind >= step) {
            swap(&data[ind - step], &data[ind]);
            ind -= step;
        }
    }
}

int main() {
    FILE * f_values;
    f_values = fopen("input.values", "r");
    if (f_values == NULL) {
        printf("Невозможно открыть файл\n");
        exit(1);
    }

    FILE * f_keys;
    f_keys = fopen("input.keys", "r");
    if (f_keys == NULL) {
        printf("Невозможно открыть файл\n");
        exit(1);
    }


    data_st data[256];
    int input_size = 0;
    while (fscanf(f_keys, "%d", &data[input_size].key) == 1) {
        fgets(data[input_size].value, 256, f_values);
        input_size++;
    }

    fclose(f_keys);
    fclose(f_values);

    printf("1.) Элементы упорядочены:\n\n");
    print_struct(data, input_size);
    printf("\n");

    printf("1.) Результат сортировки:\n\n");
    shell_sort(data, input_size, 1);
```

```
    print_struct(data, input_size);
    printf("\n");

    printf("2.) Элементы упорядочены в обратном порядке:\n\n");
    shell_sort(data, input_size, -1);
    print_struct(data, input_size);
    printf("\n");

    printf("2.) Результат сортировки:\n\n");
    shell_sort(data, input_size, 1);
    print_struct(data, input_size);
    printf("\n");

    printf("3.) Элементы не упорядочены:\n\n");
    vector_shuffle(data, input_size);
    print_struct(data, input_size);
    printf("\n");

    printf("3.) Результат сортировки:\n\n");
    shell_sort(data, input_size, 1);
    print_struct(data, input_size);
    printf("\n");
}
```

9.Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы

11. **Выводы: Узнал что такое сортировка Шелла, в процессе так же узнал про сортировку выбором и научился генерировать псевдослучайные числа.**

Подпись студента

_____