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

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

Контакты e-mail, telegram: [stas.ershov57@gmail.com](mailto: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**

1. **Цель работы: Научиться использовать сортировки**
2. **Задание** (*вариант №* **7**)**:** Сортировка Шелла
3. **Оборудование** (студента):

Процессор *Intel Pentium N4200 1.1 ГГц* с ОП *8* Гб, SSD *128* Гб. Монитор *1920x1080*

1. **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)итесты либо соображения по тестированию].

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

**#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. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы
2. **Выводы: Узнал что такое сортировка Шелла, в процессе так же узнал про сортировку выбором и научился генерировать псевдослучайные числа.**

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_