Отчет по лабораторной работе № IX по курсу
Алгоритмы и структуры данных

Студент группы М8О-103Б-22 Клименко Виталий Максимович, № по списку 11

Контакты www, e-mail, icq, skype vitalikklimenko96@gmail.com

Работа выполнена: 2 июня 2023 г.

Преподаватель: доцент Никулин С.П.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202 __ г., итоговая оценка ___

Подпись преподавателя _____

1. **Тема:** Сортировка и поиск

2. **Цель работы:** Составить программу на Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице

3. **Задание** (*вариант № 11 (11, 8)*): Быстрая сортировка Хоара рекурсивно, таблица - ключ из строки и целого, 32 байта на ключ, хранение данных и ключей вместе, минимальное число элементов - 18

4. **Оборудование** (*лабораторное*):
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:
Процессор Intel 4x 3.5GHz с ОП 16 ГБ НМД HDD 200 ГБ . Монитор Встроенный 1920x1080
Другие устройства Touchpad Synaptics

5. **Программное обеспечение** (*лабораторное*):
Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства UNIX, наименование Pop!_OS версия 22.04 jammy
интерпретатор команд bash версия 5.1.16
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных на домашнем компьютере _____

6. **Идея, метод, алгоритм** решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Создать три структуры данных - ключ, состоящая из строки, длиной 28 и целое число, элемент таблицы и сама таблица

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

Создать три файла с базами данных - отсортированный, неотсортированный и отсортированный в обратную сторону

Пункты 1-7 отчета составляются строго до начала лабораторной работы.
Допущен к выполнению работы. **Подпись преподавателя** _____

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

```
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ ls
./  ../  main.c*  makefile*  resorted.txt*  sorted.txt*  table.h*  unsorted.txt*
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ cat makefile
CC = gcc
CFLAGS = -std=c99 -Wall -Wextra -Wformat=0

main:
        $(CC) $(CFLAGS) -o main.out main.c
debug:
        $(CC) $(CFLAGS) -g -o main.out main.c

clean:
        rm -f *.o main.outvitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ cat main.c
#define TABLE_IMPLEMENTATION
#include "table.h"


int main() {
    table t = table_alloc();

    uint64_t count = 0;
    printf("Input how many rows do you want to insert: ");
    scanf("%llu", &count);

    for (uint64_t i = 0; i < count; ++i) {
        printf("Row %llu:\n", i);

        char *key_s = (char*) calloc(256, sizeof(char));
        printf("Input key string: ");
        scanf("%s", key_s);

        int key_int = 0;
        printf("Input key int: ");
        scanf("%d", &key_int);

        char *value = (char*) calloc(256, sizeof(char));
        printf("Input value string: ");
        scanf("%s", value);

        table_push(&t, key_s, key_int, value);

        free(key_s);

        printf("\n");
    }

    table_print(t);
    printf("\n");
    t = table_quick_sort(t);
    table_print(t);

    int search = 1;
    while (search) {
        char *key_s = (char*) calloc(256, sizeof(char));
        int key_int = 0;
        printf("Search for:\n");
        scanf("%s", key_s);
        scanf("%d", &key_int);

        printf("Found string: %s\n", table_binary_search(t, key_s, key_int));
        free(key_s);

        printf("Do you want to continue the search? (0/1): ");
        scanf("%d", &search);
    }

    table_dealloc(&t);
}vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ cat sorted.txt
18
zxnzmxc 123 aboba
koskj 777 comues
koskj 778 dudu
zlkjf 777 dumbela
vnatriz -50 foro
asd 1345238 g
asd 5 go
nasi 6 gz
hey -123 hello
otwe 53 ioew
```

```
ewrl[ 121 jump
soda 99 k
mai 123 labs
kek 754 lol
gain 3235 m
sodfm 2345 oooror
keyone 1 value1
key2 13 zombie
key2
13
0vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ cat unsorted.txt
18
koskj 777 comues
zxnzmxc 123 aboba
vnatriz -50 foro
koskj 778 dudu
asd 1345238 g
zlkjf 777 dumbela
nasi 6 gz
asd 5 go
hey -123 hello
soda 99 k
ewrl[ 121 jump
otwe 53 ioew
mai 123 labs
kek 754 lol
kek 112 value1
key2 13 zombie
gain 3235 m
sodfm 2345 oooror
kek
112
1
gain
3235
0vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ cat resorted.txt
18
key2 13 zombie
keyone 1 value1
sodfm 2345 oooror
gain 3235 m
kek 754 lol
mai 123 labs
soda 99 k
ewrl[ 121 jump
otwe 53 ioew
hey -123 hello
nasi 6 gz
asd 5 go
asd 1345238 g
vnatriz -50 foro
zlkjf 777 dumbela
koskj 778 dudu
koskj 777 comues
zxnzmxc 123 aboba
zxnzmxc
123
0vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ cat table.h
#ifndef TABLE_H
#define TABLE_H

#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include <stdlib.h>

#define STRING_KEY_CAP 28


typedef struct {
    char key_s[STRING_KEY_CAP];
    int key_int;
} complex_key;

typedef struct {
    complex_key key;
    char *value;
} item;

typedef struct {
    item *rows;
    uint64_t count;
} table;
```

```
table table_alloc();

table table_copy(table t);
void table_push(table *t, char *key_s, int key_int, char *value);
table table_quick_sort(table t);

char *table_binary_search(table t, char *key_s, int key_int);

void table_print(table t);

void table_dealloc(table *t);


#endif // TABLE_H



#ifdef TABLE_IMPLEMENTATION

table table_alloc() {
    table t;
    t.rows = (item*) calloc(0, sizeof(item));
    t.count = 0;
    return t;
}


table table_copy(table t) {
    table temp = table_alloc();

    for (uint64_t i = 0; i < t.count; ++i) {
        char *value = (char*) calloc(256, sizeof(char));
        strcpy(value, t.rows[i].value);
        table_push(&temp, t.rows[i].key.key_s, t.rows[i].key.key_int, value);
    }

    return temp;
}

void table_push(table *t, char *key_s, int key_int, char *value) {
    complex_key ck = {
        .key_int = key_int
    };

    for (int i = 0; i < STRING_KEY_CAP && i < (int) strlen(key_s); ++i) {
        ck.key_s[i] = key_s[i];
    }

    item i = {
        .key = ck,
        .value = value
    };

    t->rows = (item*) realloc(t->rows, sizeof(item) * (t->count + 1));
    t->count++;

    t->rows[t->count - 1] = i;
}

void _table_quick_sort(table *t, uint64_t l, uint64_t r) {
    char *pivot_value = t->rows[l].value;
    complex_key pivot_key = t->rows[l].key;
    uint64_t l_init = l, r_init = r;

    while (l < r) {
        while (
            (
                strcmp(pivot_key.key_s, t->rows[r].key.key_s) < 0 ||
                (
                    strcmp(pivot_key.key_s, t->rows[r].key.key_s) == 0 &&
                    pivot_key.key_int <= t->rows[r].key.key_int
                )
            ) && (l < r)
        ) r--;

        if (l != r) {
            t->rows[l].value = t->rows[r].value;
            t->rows[l].key = t->rows[r].key;
            l++;
        }

        while (
            strcmp(pivot_key.key_s, t->rows[l].key.key_s) > 0 &&
            (l < r)
        ) l++;
```

```
        if (l != r) {
            t->rows[r].value = t->rows[l].value;
            t->rows[r].key = t->rows[l].key;
            r--;
        }
    }

    t->rows[l].key = pivot_key;
    t->rows[l].value = pivot_value;

    uint64_t pivot = l;
    l = l_init;
    r = r_init;

    if (l < pivot) _table_quick_sort(t, l, pivot - 1);
    if (r > pivot) _table_quick_sort(t, pivot + 1, r);
}

// sort keys by elements
table table_quick_sort(table t) {
    table temp = table_copy(t);

    _table_quick_sort(&temp, 0, temp.count - 1);

    return temp;
}


char *table_binary_search(table t, char *key_s, int key_int) {
    uint64_t l = 0, r = t.count - 1, m = (l + r)/2;

    while (l <= r) {
        m = (l + r)/2;

        char *curr_key_s = t.rows[m].key.key_s;
        int curr_key_int = t.rows[m].key.key_int;

        if (strcmp(key_s, curr_key_s) == 0 && key_int == curr_key_int) {
            return t.rows[m].value;
        }

        if (strcmp(key_s, curr_key_s) > 0 ||
            (
                strcmp(key_s, curr_key_s) == 0 && key_int < curr_key_int
            )
        ) {
            l = m + 1;
        } else {
            r = m - 1;
        }
    }

    return NULL;
}


void table_print(table t) {
    printf(
        "| Key%*s| Value\n", STRING_KEY_CAP + 11 - 3, ""
    );

    for (uint64_t i = 0; i < t.count; ++i) {
        printf("| %*s%11d| %s\n", STRING_KEY_CAP, t.rows[i].key.key_s, t.rows[i].key.key_int, t.rows[i].value);
    }
}


void table_dealloc(table *t) {
    for (uint64_t i = 0; i < t->count; ++i) {
        t->rows[i].key.key_s[0] = '\0';
        t->rows[i].key.key_int = 0;
        free(t->rows[i].value);
    }

    free(t->rows);
}

#endif // TABLE_IMPLEMENTATIONvitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ make
gcc -std=c99 -Wall -Wextra -Wformat=0 -o main.out main.c
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ ./main.out < sorted.txt
Input how many rows do you want to insert: Row 0:
Input key string: Input key int: Input value string:
Row 1:
Input key string: Input key int: Input value string:
```

```
Row 2:
Input key string: Input key int: Input value string:
Row 3:
Input key string: Input key int: Input value string:
Row 4:
Input key string: Input key int: Input value string:
Row 5:
Input key string: Input key int: Input value string:
Row 6:
Input key string: Input key int: Input value string:
Row 7:
Input key string: Input key int: Input value string:
Row 8:
Input key string: Input key int: Input value string:
Row 9:
Input key string: Input key int: Input value string:
Row 10:
Input key string: Input key int: Input value string:
Row 11:
Input key string: Input key int: Input value string:
Row 12:
Input key string: Input key int: Input value string:
Row 13:
Input key string: Input key int: Input value string:
Row 14:
Input key string: Input key int: Input value string:
Row 15:
Input key string: Input key int: Input value string:
Row 16:
Input key string: Input key int: Input value string:
Row 17:
Input key string: Input key int: Input value string:
| Key                                | Value
|                       zxnzmxc        123| aboba
|                         koskj        777| comues
|                         koskj        778| dudu
|                         zlkjf        777| dumbela
|                       vnatriz        -50| foro
|                           asd    1345238| g
|                           asd          5| go
|                          nasi          6| gz
|                           hey       -123| hello
|                          otwe         53| ioew
|                         ewrl[        121| jump
|                          soda         99| k
|                           mai        123| labs
|                           kek        754| lol
|                          gain       3235| m
|                         sodfm       2345| oooror
|                        keyone          1| value1
|                          key2         13| zombie

| Key                                | Value
|                           asd          5| go
|                           asd    1345238| g
|                         ewrl[        121| jump
|                          gain       3235| m
|                           hey       -123| hello
|                           kek        754| lol
|                          key2         13| zombie
|                        keyone          1| value1
|                         koskj        777| comues
|                         koskj        778| dudu
|                           mai        123| labs
|                          nasi          6| gz
|                          otwe         53| ioew
|                          soda         99| k
|                         sodfm       2345| oooror
|                       vnatriz        -50| foro
|                         zlkjf        777| dumbela
|                       zxnzmxc        123| aboba
Search for:
Found string: zombie
Do you want to continue the search? (0/1): vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ ./main.out < resorted.txt
Input how many rows do you want to insert: Row 0:
Input key string: Input key int: Input value string:
Row 1:
Input key string: Input key int: Input value string:
Row 2:
Input key string: Input key int: Input value string:
Row 3:
Input key string: Input key int: Input value string:
Row 4:
Input key string: Input key int: Input value string:
Row 5:
```

```
Input key string: Input key int: Input value string:
Row 6:
Input key string: Input key int: Input value string:
Row 7:
Input key string: Input key int: Input value string:
Row 8:
Input key string: Input key int: Input value string:
Row 9:
Input key string: Input key int: Input value string:
Row 10:
Input key string: Input key int: Input value string:
Row 11:
Input key string: Input key int: Input value string:
Row 12:
Input key string: Input key int: Input value string:
Row 13:
Input key string: Input key int: Input value string:
Row 14:
Input key string: Input key int: Input value string:
Row 15:
Input key string: Input key int: Input value string:
Row 16:
Input key string: Input key int: Input value string:
Row 17:
Input key string: Input key int: Input value string:
| Key                                    | Value
|                          key2         13| zombie
|                        keyone          1| value1
|                         sodfm       2345| oooror
|                          gain       3235| m
|                           kek        754| lol
|                           mai        123| labs
|                          soda         99| k
|                         ewrl[        121| jump
|                          otwe         53| ioew
|                           hey       -123| hello
|                          nasi          6| gz
|                           asd          5| go
|                           asd    1345238| g
|                       vnatriz        -50| foro
|                         zlkjf        777| dumbela
|                         koskj        778| dudu
|                         koskj        777| comues
|                       zxnzmxc        123| aboba

| Key                                    | Value
|                           asd          5| go
|                           asd    1345238| g
|                         ewrl[        121| jump
|                          gain       3235| m
|                           hey       -123| hello
|                           kek        754| lol
|                          key2         13| zombie
|                        keyone          1| value1
|                         koskj        777| comues
|                         koskj        778| dudu
|                           mai        123| labs
|                          nasi          6| gz
|                          otwe         53| ioew
|                          soda         99| k
|                         sodfm       2345| oooror
|                       vnatriz        -50| foro
|                         zlkjf        777| dumbela
|                       zxnzmxc        123| aboba
Search for:
Found string: aboba
Do you want to continue the search? (0/1): vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$ ./main.out < unsorted.txt
Input how many rows do you want to insert: Row 0:
Input key string: Input key int: Input value string:
Row 1:
Input key string: Input key int: Input value string:
Row 2:
Input key string: Input key int: Input value string:
Row 3:
Input key string: Input key int: Input value string:
Row 4:
Input key string: Input key int: Input value string:
Row 5:
Input key string: Input key int: Input value string:
Row 6:
Input key string: Input key int: Input value string:
Row 7:
Input key string: Input key int: Input value string:
Row 8:
Input key string: Input key int: Input value string:
```

```
Row 9:
Input key string: Input key int: Input value string:
Row 10:
Input key string: Input key int: Input value string:
Row 11:
Input key string: Input key int: Input value string:
Row 12:
Input key string: Input key int: Input value string:
Row 13:
Input key string: Input key int: Input value string:
Row 14:
Input key string: Input key int: Input value string:
Row 15:
Input key string: Input key int: Input value string:
Row 16:
Input key string: Input key int: Input value string:
Row 17:
Input key string: Input key int: Input value string:
| Key                                 | Value
|                         koskj       777| comues
|                        zxnzmxc      123| aboba
|                        vnatriz      -50| foro
|                          koskj      778| dudu
|                            asd  1345238| g
|                          zlkjf      777| dumbela
|                           nasi        6| gz
|                            asd        5| go
|                            hey     -123| hello
|                           soda       99| k
|                          ewrl[      121| jump
|                           otwe       53| ioew
|                            mai      123| labs
|                            kek      754| lol
|                            kek      112| value1
|                           key2       13| zombie
|                           gain     3235| m
|                          sodfm     2345| oooror

| Key                                 | Value
|                            asd        5| go
|                            asd  1345238| g
|                          ewrl[      121| jump
|                           gain     3235| m
|                            hey     -123| hello
|                            kek      754| lol
|                            kek      112| value1
|                           key2       13| zombie
|                          koskj      777| comues
|                          koskj      778| dudu
|                            mai      123| labs
|                           nasi        6| gz
|                           otwe       53| ioew
|                           soda       99| k
|                          sodfm     2345| oooror
|                        vnatriz      -50| foro
|                          zlkjf      777| dumbela
|                        zxnzmxc      123| aboba
Search for:
Found string: value1
Do you want to continue the search? (0/1): Search for:
Found string: m
Do you want to continue the search? (0/1): vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/lix$
```

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 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:** Я научился сортировать таблицу, искать в ней значения при помощи бинарного поиска

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____