

Отчет по лабораторной работе № IX

по курсу: 1 фундаментальная информатика

студент группы : М8О-105Б-21 Титеев Рамиль Маратович , № по списку: 23

Адреса www, e-mail, jabber, skype: derol.gym@gmail.com

Работа выполнена: "13 мая 2022г"

Преподаватель: каф. 806 В.К.Титов

Входной контроль знаний с оценкой: _____

Отчет сдан "___" _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Сортировка и поиск.

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

3. Задание: Пирамидальная сортировка с просеиванием;

4. Оборудование(лабораторное):

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ ГБ

НМД _____ ГБ. Терминал _____ адрес _____. Принтер _____

Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Ryzen 4600 @ 6x 3.0 GHz , ОП 16384 МБ, НМД _____ ГБ. Монитор Встроенный

Другие устройства _____

5. Программное обеспечение(лабораторное):

Операционная система семейства UNIX, наименование _____ версия _____

Интерпретатор команд: _____ версия _____

Система программирования: _____ версия _____

Редактор текстов: _____ версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства UNIX, наименование Ubuntu версия 20.04

Интерпретатор команд: bash версия _____

Система программирования: C версия _____

Редактор текстов: Emacs версия _____

Утилиты операционной системы: _____

Прикладные системы и программы: _____

Местонахождение и имена файлов и программ данных: /usr/bin , а также /bin

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

Алгоритм сортировки заключается в следующем:

Для начала формируется нижний ряд пирамиды. Для этого в цикле пока он не дойдет до последнего ряда каждую итерацию делаем проверку, если мы оказались в последнем ряду, то запоминаем запоминаем левый потомок, иначе запоминаем наибольший потомок из двух. Затем проверяется, если элемент вершины меньше максимального потомка, то меняем их местами, иначе выходим из цикла.

Затем просеиваем через пирамиду остальные элементы таким же алгоритмом.

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

curs_9.cpp:

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

int is_sorted = 0;

struct Row{
    int key;
    char *string;
};

int get_size(FILE *input){
    int n = 0;
    char* buffer = new char[128];
    size_t l = sizeof(buffer);
    while(!feof(input)){
        getline(&buffer, &l, input);
        n++;
    }
    delete [] buffer;
    rewind(input);
    return n;
}

void print_table(Row *table, int size){
    printf("-----\n");
    printf("| Key |                               Data\n");
    printf("      |\n");
    printf("-----\n");
    for(int i = 0; i<size; ++i){
        printf("%3d | %s", table[i].key, table[i].string);
        printf("-----\n");
    }
    printf("\n");
}

Row * make_table(FILE *keys, FILE *data, int size){
    Row * table = new Row[size];
```

```

    for (int i = 0; i < size; ++i){
        table[i].string = new char[128];
    }
    for (int i = 0; i < size; ++i){
        fscanf(keys, "%d", &table[i].key);
        fgets(table[i].string, 128, data);
    }
    return table;
}

void swap_strings(Row * table, int a, int b){
    Row tmp;
    tmp = table[a];
    table[a] = table[b];
    table[b] = tmp;
}

void siftDown(Row *table, int root, int bottom){
    int maxChild;
    int done = 0;
    while ((root*2 <= bottom) && (!done)){
        if (root*2 == bottom)
            maxChild = root*2;
        else if (table[root*2].key > table[root*2+1].key){
            maxChild = root*2;
        }
        else{
            maxChild = root*2+1;
        }
        if (table[root].key < table[maxChild].key){
            swap_strings(table, root, maxChild);
            root = maxChild;
        }
        else done = 1;
    }
}

void heapSort(Row *table, int size){
    for (int i = (size / 2); i >= 0; i--){
        siftDown(table, i, size - 1);
    }
    for (int i = size - 1; i >= 1; i--){
        swap_strings(table, 0, i);
        siftDown(table, 0, i - 1);
    }
}

void sort_table(Row * table, int size){
    is_sorted = 1;
    heapSort(table, size);
}

void mix_table(Row * table, int size){
    is_sorted = 0;

```

```

    int a, b;
    for(int i = 0; i<size; ++i){
        a = rand() % size;
        b = rand() % size;
        swap_strings(table, a, b);
    }
}

int bin_search(Row * table, int key, int size){
    int l = 0, r = size - 1, mid;
    while (l<=r){
        mid = (r + l)/2;
        if (key == table[mid].key){
            return mid;
        }
        else if(key < table[mid].key){
            r = mid;
        }
        else if(key > table[mid].key){
            l = mid+1;
        }
    }
    return -1;
}

void reverse_strings(Row * table, int size){
    is_sorted = 0;
    for(int i = 0, j = size - 1; i < j; ++i, --j){
        swap_strings(table, i, j);
    }
}

int main(){
    srand(time(0));
    FILE* file1 = fopen("in1", "r");
    if (file1 == NULL ){
        printf("Can not open file1\n");
        return 0;
    }
    FILE* file2 = fopen("in2", "r");
    if (file2 == NULL ){
        printf("Can not open file2\n");
        return 0;
    }
    int n = get_size(file1), action;
    Row *table = make_table(file1, file2, n);
    fclose(file1);
    fclose(file2);
    print_table(table, n);
    while(1){
        printf("Menu\n");
        printf("1) Print table\n");
        printf("2) Binary search\n");
        printf("3) Sort table\n");
    }
}

```

```

printf("4) Mix table\n");
printf("5) Reverse table\n");
printf("6) Exit\n");
printf("Choose an action ==> ");
scanf("%d", &action);
switch (action){
    case 1:{
        print_table(table, n);
        break;
    }
    case 2:{
        if(is_sorted){
            printf("Enter the key: ");
            int k;
            scanf("%d", &k);
            int search;
            printf("\n");
            search = bin_search(table, k, n);
            if (search == -1){
                printf("Element with such key is not found!\n");
                break;
            }
            printf("Found the string:\n %s", table[search].string);
        }
        else printf("Table is not sorted!\n");
        break;
    }
    case 3:{
        sort_table(table, n);
        break;
    }
    case 4:{
        mix_table(table, n);
        break;
    }
    case 5:{
        reverse_strings(table, n);
        break;
    }
    case 6: return 0;
}
}
}

```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

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

```

constantfear@constantfear:~/projects/laboratory/2_semester/curs_9$ cat header.txt
*****
*                               *
*           Курсовая работа №9   *
*           Сортировка и поиск.   *
*           Выполнил студент гр. М80-105-Б *
*                               *

```

```

*                               Титеев Рамиль Маратович                               *
*****
constantfear@constantfear:~/projects/laboratory/2_semester/curs_9$ cat in1
1
2
3
4
5
6
7
8
constantfear@constantfear:~/projects/laboratory/2_semester/curs_9$ cat in2
Morbi mattis massa ac elit pulvinar dictum.
In in justo id diam pellentesque congue.
Donec vitae tellus non erat egestas scelerisque.
Quisque quis est et leo sollicitudin dapibus.
Nam rutrum diam sed tincidunt auctor.
Duis lobortis lacus nec euismod mattis.
Donec ut elit cursus, ornare nisi a, mollis mi.
Aliquam blandit nulla quis ultrices volutpat.
constantfear@constantfear:~/projects/laboratory/2_semester/curs_9$ g++ curs_9.cpp
constantfear@constantfear:~/projects/laboratory/2_semester/curs_9$ ./a.out

```

Key	Data
1	Morbi mattis massa ac elit pulvinar dictum.
2	In in justo id diam pellentesque congue.
3	Donec vitae tellus non erat egestas scelerisque.
4	Quisque quis est et leo sollicitudin dapibus.
5	Nam rutrum diam sed tincidunt auctor.
6	Duis lobortis lacus nec euismod mattis.
7	Donec ut elit cursus, ornare nisi a, mollis mi.
8	Aliquam blandit nulla quis ultrices volutpat.

Menu

- 1) Print table
- 2) Binary search
- 3) Sort table
- 4) Mix table
- 5) Reverse table
- 6) Exit

Choose an action ==> 4

Menu

- 1) Print table
- 2) Binary search
- 3) Sort table
- 4) Mix table
- 5) Reverse table
- 6) Exit

Choose an action ==> 1

Key	Data
1	Morbi mattis massa ac elit pulvinar dictum.

3	Donec vitae tellus non erat egestas scelerisque.
4	Quisque quis est et leo sollicitudin dapibus.
7	Donec ut elit cursus, ornare nisi a, mollis mi.
5	Nam rutrum diam sed tincidunt auctor.
6	Duis lobortis lacus nec euismod mattis.
8	Aliquam blandit nulla quis ultrices volutpat.
2	In in justo id diam pellentesque congue.

Menu

- 1) Print table
- 2) Binary search
- 3) Sort table
- 4) Mix table
- 5) Reverse table
- 6) Exit

Choose an action ==> 2

Table is not sorted!

Menu

- 1) Print table
- 2) Binary search
- 3) Sort table
- 4) Mix table
- 5) Reverse table
- 6) Exit

Choose an action ==> 5

Menu

- 1) Print table
- 2) Binary search
- 3) Sort table
- 4) Mix table
- 5) Reverse table
- 6) Exit

Choose an action ==> 1

Key	Data
2	In in justo id diam pellentesque congue.
8	Aliquam blandit nulla quis ultrices volutpat.
6	Duis lobortis lacus nec euismod mattis.
5	Nam rutrum diam sed tincidunt auctor.
7	Donec ut elit cursus, ornare nisi a, mollis mi.
4	Quisque quis est et leo sollicitudin dapibus.
3	Donec vitae tellus non erat egestas scelerisque.
1	Morbi mattis massa ac elit pulvinar dictum.

Menu

- 1) Print table
- 2) Binary search

```

3) Sort table
4) Mix table
5) Reverse table
6) Exit
Choose an action ==> 3
Menu
1) Print table
2) Binary search
3) Sort table
4) Mix table
5) Reverse table
6) Exit
Choose an action ==> 1

```

Key	Data
1	Morbi mattis massa ac elit pulvinar dictum.
2	In in justo id diam pellentesque congue.
3	Donec vitae tellus non erat egestas scelerisque.
4	Quisque quis est et leo sollicitudin dapibus.
5	Nam rutrum diam sed tincidunt auctor.
6	Duis lobortis lacus nec euismod mattis.
7	Donec ut elit cursus, ornare nisi a, mollis mi.
8	Aliquam blandit nulla quis ultrices volutpat.

```

Menu
1) Print table
2) Binary search
3) Sort table
4) Mix table
5) Reverse table
6) Exit
Choose an action ==> 2
Enter the key: 6

```

```

Found the string:
Duis lobortis lacus nec euismod mattis.

```

```

Menu
1) Print table
2) Binary search
3) Sort table
4) Mix table
5) Reverse table
6) Exit
Choose an action ==> 6

```

9. Дневник отладки должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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. Замечание автора по существу работы _____

11. Выводы _____ Я научился работать с линейными списками.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента _____