Московский авиационный институт

(Национальный исследовательский университет)

Институт прикладной математики и информатики

Кафедра вычислительной математики и программирования

**Задание 9 по курсовому проекту**

**«Сортировка и поиск»**

Руководитель: Титов В.К.

\_\_\_5\_\_\_\_

(оценка)

\_\_ВК Т\_\_\_ 2 июня 2020 г.

(подпись) (дата)

Студент группы М8О-10ХБ-19:

Москва 2020

Оглавление:

[1. Задание 3](#_Toc483737568)

[2. Общий метод решения 3](#_Toc483737569)

[3. Общее сведения о программе 4](#_Toc483737570)

[4. Функциональное назначение 4](#_Toc483737571)

[5. Логическая структура 4](#_Toc483737572)

[6. Текст программы](#_Toc483737573) 4

[7. Описание переменных](#_Toc483737574) 8

[8. Описание функций](#_Toc483737574) 9

[9. Входные данные](#_Toc483737575) 9

[10. Выходные данные 9](#_Toc483737576)

[11. Тестирование 10](#_Toc483737577)

[12. Журнал отладки 13](#_Toc483737578)

[13. Вывод по задаче 1](#_Toc483737579)3

1. Задание

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

Программа должна вводить значения элементов неупорядоченной таблицы и проверять работу процедуры сортировки в трех случаях:

1. элементы таблицы с самого начала упорядочены;
2. элементы таблицы расставлены в обратном порядке;
3. элементы таблицы не упорядочены.

В последнем случае можно использовать встроенные процедуры генерации псевдослучайных чисел.

Для каждого вызова процедуры сортировки необходимо печатать исходное состояние таблицы и результаты сортировки. После выполнения сортировки программа должна вводить ключи и для каждого из них выполнять поиск в

упорядоченной таблице с помощью процедуры двоичного поиска и печатать найденные элементы, если они присутствуют в таблице.

Вариант N=20.

Метод сортировки: Быстрая сортировка Хоара (рекурсивный вариант).

Тип ключа: комплексный, хранение вместе с данными.

1. Общий метод решения

Быстрая сортировка является существенно улучшенным вариантом сортировки с помощью прямого обмена. Суть её предельно проста: выбирается так называемый опорный или зерновой элемент, и массив делится на 2 подмассива: меньших опорного и равных или больших опорного. Потом этот алгоритм применяется рекурсивно к подмассивам. Опорный элемент будем выбирать как средний по индексу в массиве. Если сократить и подытожить все, то получим следующий алгоритм:

1. Выбрать опорный элемент из массива.
2. Перераспределить элементы массива таким образом, чтобы элементы меньше опорного были перед ним, а большие или равные после него.
3. Рекурсивно применить два шага к двум подмассивам слева и справа опорного элемента. Рекурсия не применяется к массиву, в котором меньше двух элементов.

Бинарный поиск – классический алгоритм поиска элемента в отсортированном массиве, использующий дробление массива на половины. Алгоритм:

1. Определение значения элемента в середине структуры данных. Полученное значение сравнивается с ключом.
2. Если ключ меньше значения середины, то поиск осуществляется в первой половине элементов, иначе — во второй.
3. Поиск сводится к тому, что вновь определяется значение серединного элемента в выбранной половине и сравнивается с ключом.
4. Процесс продолжается до тех пор, пока не будет найден элемент со значением ключа или не станет пустым интервал для поиска.

Комплексные числа – числа, это числа вида где – вещественные числа, а мнимая единица, т.е. число, для которого выполняется равенство . Для сравнения ключей будем сравнивать квадраты вещественных чисел комплексного числа. Т.е. , если .

1. Общие сведения о программе

В связи с дистанционным обучением, тестирование, отладка и протоколирование программ производилось на домашнем ПК.

Система программирования: GNU C/C++.

Местонахождение файлов: домашний компьютер с OC Windows, работа в Windows Subsystem Linux.

kng@Legion:/mnt/c/vsc/labs$

Способ компиляции: gcc kurs9.cpp -o kurs9.exe

1. Функциональное назначение

Программа предназначена для демонстрации сортировки методом Хоара, а также для демонстрации работы бинарного поиска.

1. Логическая структура

Перед основной функцией объявлены функции: get\_size, print\_table, make\_table, to\_value, sort\_table, swap\_strings, mix\_table, bin\_search, reverse\_strings, а также структуры Key и PoetryString. Структура Key содержит в себе два значения: значение действительной части ключа, и значение мнимой. Структура PoetryString содержит ключ типа Key, а также символьную строку.

1. Текст программы

#include <stdio.h>

#include <stdlib.h>

int is\_sorted = 0;

struct Key{

    short im;

    short re;

};

struct PoetryString{

    Key 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(PoetryString \*table, int size){

    for(int i = 0; i < size; ++i){

        printf("%3hi %2hi ----- %s", table[i].key.re, table[i].key.im, table[i].string);

    }

    printf("\n");

}

PoetryString \* make\_table(FILE \*input, int size){

    PoetryString \*table = new PoetryString [size];

    for (int i = 0; i < size; ++i){

        table[i].string = new char[128];

    }

    for(int i = 0; i < size; ++i){

        fscanf(input, "%hi %hi", &table[i].key.re, &table[i].key.im);

        fgets(table[i].string, 128, input);

    }

    return table;

}

int to\_value(Key k){

    return (k.re\*k.re + k.im\*k.im);

}

void sort\_table(PoetryString \* table, int left, int right){

    is\_sorted = 1;

    int l = left, r = right, mid;

    PoetryString temp;

    mid = to\_value(table[(l+r)/2].key);

    do{

        while(mid > to\_value(table[l].key)) l++;

        while(mid < to\_value(table[r].key)) r--;

        if( l <= r) {

            temp = table[l];

            table[l++] = table[r];

            table[r--] = temp;

        }

    } while (l <= r);

    if (left < r) sort\_table(table, left, r);

    if (l < right) sort\_table(table, l, right);

}

void swap\_strings(PoetryString\* table, int a, int b){

    PoetryString temp;

    temp = table[a];

    table[a] = table[b];

    table[b] = temp;

}

void mix\_table(PoetryString \*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(PoetryString \*table, Key key, int size){

    int l = 0, r = size - 1, mid;

    while (l <= r){

        mid = (r + l)/2;

        if(key.re == table[mid].key.re && key.im == table[mid].key.im) return mid;

        if(to\_value(key) < to\_value(table[mid].key)) r = mid;

        if(to\_value(key) > to\_value(table[mid].key)) l = mid + 1;

    }

    return -1;

}

void reverse\_strings(PoetryString \*table, int size){

    is\_sorted = 0;

    for(int i = 0, j = size - 1; i < j; ++i, --j){

        swap\_strings(table,i, j);

    }

}

int main(){

    FILE \*input;

    if ((input = fopen("poetry.txt", "r")) == NULL){

        printf("Error: can't input from poetry.txt\n");

        return 1;

    }

    int n = get\_size(input), action;

    n--;

    PoetryString \*table = make\_table(input, n);

    print\_table(table, n);

    while(1){

        printf("Menu\n");

        printf("1) Binary search\n");

        printf("2) Sort\n");

        printf("3) Mix\n");

        printf("4) Reverse\n");

        printf("5) Exit\n");

        printf("Choose an action\n");

        scanf("%d", &action);

        switch (action){

            case 1:{

                if(is\_sorted){

                    printf("Enter the real and imaginary parts of the key: ");

                    Key k;

                    scanf("%hi %hi", &k.re, &k.im);

                    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 2:{

                sort\_table(table, 0, n);

                print\_table(table, n);

                break;

            }

            case 3:{

                mix\_table(table, n);

                print\_table(table,n);

                break;

            }

            case 4:{

                reverse\_strings(table, n);

                print\_table(table,n);

                break;

            }

            case 5: return 0;

        }

    }

}

1. Описание переменных

**Key**

|  |  |  |
| --- | --- | --- |
| Переменная | Тип | Данные |
| re | short int | Значение вещественной части комплексного числа |
| im | short int | Коэффициент перед мнимой единицей |

**PoetryString**

|  |  |  |
| --- | --- | --- |
| Переменная | Тип | Данные |
| key | Key | Комплексный ключ строки |
| string | char\* | Строка стихотворения |

|  |  |  |
| --- | --- | --- |
| Переменная | Тип | Данные |
| is\_sorted | int | Глобальная переменная, используемая для запоминания, отсортирована ли таблица, или нет |
| buffer | char\* | Переменная, в которую считываются строки, при подсчете количества строк |
| l | size\_t | Переменная, равная размеру переменной buffer. Нужна для функции getline. Встречается и в другой функции, с другим типом и назначением (см. ниже) |
| n | int | Переменная, хранящая в себе количество строк в таблице. |
| l | int | Переменная, используемая для сортировки и бинарного поиска. Хранит в себе левую границу обрабатываемой части массива. |
| r | int | Переменная, используемая для сортировки и бинарного поиска. Хранит в себе правую границу обрабатываемой части массива. |
| mid | int | Переменная, используемая для сортировки и бинарного поиска. Хранит в себе индекс середины обрабатываемого массива. |
| temp | PoetryString | Переменная, используемая для сортировки и обмена значений элементов таблицы. |
| table | PoetryString\* | Таблица, в которой хранятся ключи и строки исходного стихотворения |
| a | int | Переменная с индексом строки, используется для обмена значений элементов таблицы и для перемешивания таблицы |
| b | int | Переменная с индексом строки, используется для обмена значений элементов таблицы и для перемешивания таблицы |
| input | FILE \* | Указатель на файл со стихотворением |

1. Описание функций

|  |  |  |
| --- | --- | --- |
| Функция | Тип | Назначение |
| get\_size | int | Считает количество строк в стихотворении |
| print\_table | void | Печатает таблицу |
| make\_table | PoetryString\* | Строит таблицу по стихотворению, каждому элементу таблицы соответствует ключ и строка стихотворения |
| to\_value | int | Преобразует значения ключа в сумму квадратов коэффициентов, для того чтобы сравнивать разные комплексные ключи |
| sort\_table | void | Сортирует таблицу методом Хоара |
| swap\_strings | void | Обменивает значения двух заданных элементов таблицы |
| mix\_table | void | Смешивает строки таблицы в случайном порядке |
| bin\_search | int | Ищет заданный ключ в таблице бинарным поиском |
| reverse\_strings | void | Переворачивает порядок строк |

1. Входные данные

Работа с программой ведется в консоли из меню программы.

Для тестирования предоставлен фрагмент стихотворения Роберта Ли Фроста – “The Road Not Taken”:

100 10 Two roads diverged in a yellow wood,

101 11 And sorry I could not travel both

102 12 And be one traveler, long I stood

103 13 And looked down one as far as I could

104 14 To where it bent in the undergrowth.

105 15 Then took the other, as just as fair,

106 16 And having perhaps the better claim,

107 17 Because it was grassy and wanted wear;

108 18 Though as for that the passing there

109 19 Had worn them really about the same.

1. Выходные данные

Выходные данные выводятся в консоль в результате работы с меню программы.

1. Тестирование

kng@Legion:/mnt/c/vsc/labs$ cat head.txt

▇▇▇▇▇▇▇▇▇▇▇▇▓▒░• ۵ •░▒▓▇▇▇▇▇▇▇▇▇▇▇▇

▇Курсовая работа №9 ▇

▇Сортировка и поиск ▇

▇Выполнил студент группы 104 ▇

▇Тарпанов Даниил Александрович ▇

▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇

kng@Legion:/mnt/c/vsc/labs$ gcc kurs9.cpp -o kurs9.exe

kng@Legion:/mnt/c/vsc/labs$ cat poetry.txt

100 10 Two roads diverged in a yellow wood,

101 11 And sorry I could not travel both

102 12 And be one traveler, long I stood

103 13 And looked down one as far as I could

104 14 To where it bent in the undergrowth.

105 15 Then took the other, as just as fair,

106 16 And having perhaps the better claim,

107 17 Because it was grassy and wanted wear;

108 18 Though as for that the passing there

109 19 Had worn them really about the same.

kng@Legion:/mnt/c/vsc/labs$ ./kurs9.exe

100 10 ----- Two roads diverged in a yellow wood,

101 11 ----- And sorry I could not travel both

102 12 ----- And be one traveler, long I stood

103 13 ----- And looked down one as far as I could

104 14 ----- To where it bent in the undergrowth.

105 15 ----- Then took the other, as just as fair,

106 16 ----- And having perhaps the better claim,

107 17 ----- Because it was grassy and wanted wear;

108 18 ----- Though as for that the passing there

109 19 ----- Had worn them really about the same.

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

3

104 14 ----- To where it bent in the undergrowth.

101 11 ----- And sorry I could not travel both

105 15 ----- Then took the other, as just as fair,

103 13 ----- And looked down one as far as I could

102 12 ----- And be one traveler, long I stood

109 19 ----- Had worn them really about the same.

107 17 ----- Because it was grassy and wanted wear;

106 16 ----- And having perhaps the better claim,

108 18 ----- Though as for that the passing there

100 10 ----- Two roads diverged in a yellow wood,

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

1

Table is not sorted!

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

2

100 10 ----- Two roads diverged in a yellow wood,

101 11 ----- And sorry I could not travel both

102 12 ----- And be one traveler, long I stood

103 13 ----- And looked down one as far as I could

104 14 ----- To where it bent in the undergrowth.

105 15 ----- Then took the other, as just as fair,

106 16 ----- And having perhaps the better claim,

107 17 ----- Because it was grassy and wanted wear;

108 18 ----- Though as for that the passing there

109 19 ----- Had worn them really about the same.

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

4

109 19 ----- Had worn them really about the same.

108 18 ----- Though as for that the passing there

107 17 ----- Because it was grassy and wanted wear;

106 16 ----- And having perhaps the better claim,

105 15 ----- Then took the other, as just as fair,

104 14 ----- To where it bent in the undergrowth.

103 13 ----- And looked down one as far as I could

102 12 ----- And be one traveler, long I stood

101 11 ----- And sorry I could not travel both

100 10 ----- Two roads diverged in a yellow wood,

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

3

109 19 ----- Had worn them really about the same.

104 14 ----- To where it bent in the undergrowth.

100 10 ----- Two roads diverged in a yellow wood,

107 17 ----- Because it was grassy and wanted wear;

101 11 ----- And sorry I could not travel both

108 18 ----- Though as for that the passing there

102 12 ----- And be one traveler, long I stood

106 16 ----- And having perhaps the better claim,

103 13 ----- And looked down one as far as I could

105 15 ----- Then took the other, as just as fair,

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

2

100 10 ----- Two roads diverged in a yellow wood,

101 11 ----- And sorry I could not travel both

102 12 ----- And be one traveler, long I stood

103 13 ----- And looked down one as far as I could

104 14 ----- To where it bent in the undergrowth.

105 15 ----- Then took the other, as just as fair,

106 16 ----- And having perhaps the better claim,

107 17 ----- Because it was grassy and wanted wear;

108 18 ----- Though as for that the passing there

109 19 ----- Had worn them really about the same.

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

1

Enter the real and imaginary parts of the key: 106 16

Found the string:

And having perhaps the better claim,

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

1

Enter the real and imaginary parts of the key: 120 30

Element with such key is not found!

Menu

1) Binary search

2) Sort

3) Mix

4) Reverse

5) Exit

Choose an action

5

kng@Legion:/mnt/c/vsc/labs$

1. Журнал отладки

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Время** | **Проблема** | **Решение** |
| 1.06.20 |  | Некорректное считывание | Сначала считывать ключ через fscanf, а потом оставшуюся строку через fgets |

1. Вывод по задаче

Изучил быструю сортировку. Потренировался в работе с потоками ввода, считывании нескольких типов данных из файла. Реализовал структуру и функцию для работы с комплексными числами. Еще раз потренировался в работе с памятью.