Московский Авиационный Институт

*(Национальный Исследовательский Университет)*

**Факультет №8 «Информационные технологии и прикладная математика»**

**Кафедра 806 «Вычислительная математика и программирование»**

Курсовой проект по предмету

*«Языки и методы программирования»*

2 семестр

Задание №9

*«Сортировка и поиск».*

|  |  |
| --- | --- |
| Студент: |  |
| Группа: | М8О-10ХБ-19 |
| Преподаватель: | Титов В.К. |
| Оценка: | 5 |
| Дата: | 1.06.2020 |

Подпись: ВК Т\_

Москва

*2020*

**Содержание**

Задание……………………………………………………………………………………………….3

Теоретические сведения……….…………………………………………………………..3

Описание программы…………………..…….………………………………………………4

Взаимодействие с программой.…………………………………………………………4

Описание переменных и функций………………………………………………….….5

Сценарий…..………………………………………………………………………………..………6

Протокол....………………………………………………………………………………..………10

Заключение………………………………………………………………………………………..19

**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

• Элементы таблицы с самого начала упорядочены.

• Элементы таблицы расставлены в обратном порядке.

• Элементы таблицы не упорядочены.

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.

**Метод сортировки:** линейный выбор с подсчетом (вариант 2)

**Тип ключа:** комплексный (вариант 9)

**Хранение данных и ключей:** вместе

**Текст для таблицы:** фрагмент из стихотворения

**Теоретические сведения**

**Двоичный (бинарный) поиск**) — классический алгоритм поиска элемента в отсортированном массиве (векторе), использующий дробление массива на половины. Используется в информатике, вычислительной математике и математическом программировании.

**Сортировка подсчётом** — алгоритм сортировки, в котором используется диапазон чисел сортируемого массива (списка) для подсчёта совпадающих элементов. Применение сортировки подсчётом целесообразно лишь тогда, когда сортируемые числа имеют (или их можно отобразить в) диапазон возможных значений, который достаточно мал по сравнению с сортируемым множеством, например, миллион натуральных чисел меньших 1000.

Для сортировки элементов исходного массива А размерности n, нужно сформировать массив счетчиков S размерности n, каждый элемент которого будет показывать, в какой позиции должен стоять соответствующий элемент массива А в упорядоченном списке.

На первом этапе в массив счетчиков заносятся нули.

При первом просмотре для первого элемента массива A подсчитывается количество меньших элементов во всем массиве и это количество заносится в счетчик первого элемента, в счетчики элементов, больших первого, добавляются единицы.

При втором просмотре первый элемент массива A исключается из рассмотрения и для второго элемента в оставшемся массиве подсчитывается количество меньших элементов (это количество заносится в счетчик второго элемента), в счетчики больших элементов добавляются единицы, и.т.д.

Перемещая элементы исходного массива A в полученный массив B в соответствии со значениями массива счетчиков S, получим упорядоченный исходный массив B.

**Описание программы**

Программа состоит из одного файл kp9.c.

Система программирования: GNU C/C++.

Компиляция: gcc kp9.c –o kp9.exe

Запуск: ./kp9.exe

**Взаимодействие с программой**

После запуска программы, на экран пользователя выводится список доступных команд:

• print – вывести таблицу.

• binary search – выполняет поиск в таблице по заданному ключу (необходимо ввести действительную и мнимую часть ключа)

• sort – сортирует таблицу по возрастанию ключей.

• mix – перемешивает элементы списка в случайном порядке.

• reverse – располагает элементы таблицы в обратном порядке.

• quit – выход из программы.

**Описание используемых переменных и функций**

**Структурный тип Key**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| a | short int | Действительная часть |
| b | short int | Мнимая часть |

**Структурный тип Row**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| key | Key | Ключ |
| val[] | char | Значение элемента |

**Переменные и функции**

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| N | const int | Максимальное число элементов таблицы |
| data[N] | Row | Таблица элементов |
| key | Key | Ключ |
| i | int | Счетчик |
| cnt | int | Количество введенных элементов |
| action | int | Выбор пункта меню |
| printable() | void | Печать таблицы |
| binSearch() | int | Бинарный поиск |
| sort() | void | Сортировка |
| scramble() | void | Перемешивание элементов |
| reverse() | void | Обратный порядок элементов |
| comparator() | int | Сравнение ключей |
| isSorted() | int | Проверка, отсортирована ли таблица |
| swapRows() | void | Поменять элементы местами |  |  |
| getRow() | void | Ввод строки |  |  |
| randomAB() | int | Вывод случайного значения |  |  |

**Сценарий программы**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

typedef struct \_Key

{

short int a;

short int b;

} Key;

typedef struct \_Row

{

Key key;

char val[80];

} Row;

void printTable(const Row\* rows, const int size);

int binSearch(const Row\* rows, const int size, const Key key);

void sort(Row\* rows, const int size);

void scramble(Row\* rows, const int size);

void reverse(Row\* rows, const int size);

void getRow(FILE\* stream, char\* str, const int size);

void swapRows(Row\* rows, const int a, const int b);

int comparator(const Key k1, const Key k2);

int randomAB(const int a, const int b);

int isSorted(const Row\* rows, const int size);

void printTable(const Row\* rows, const int size)

{

int i;

printf("+---------+-----------------------------------------------------------------------------+\n");

printf("| Key | string |\n");

printf("+---------+-----------------------------------------------------------------------------+\n");

for (i = 0; i < size; i++)

printf("|%4hd %4hd|%70s\n", rows[i].key.a, rows[i].key.b, rows[i].val);

printf("+---------+-----------------------------------------------------------------------------+\n");

}

int binSearch(const Row\* rows, const int size, const Key key)

{

int start = 0, end = size - 1, mid;

if (size <= 0)

return -1;

while (start < end)

{

mid = start + (end - start) / 2;

if (comparator(rows[mid].key, key) == 0)

return mid;

else if (comparator(rows[mid].key, key) < 1)

start = mid + 1;

else

end = mid;

}

if (comparator(rows[end].key, key) == 0)

return end;

return -1;

}

void sort(Row\* rows, const int size)

{

int i, j;

int count[size];

Row b[size];

for (i = 0; i < size; i++)

count[i] = 0;

for (i = 0; i < size - 1; i++)

for (j = i + 1; j < size; j++)

if (comparator(rows[i].key, rows[j].key) < 1)

count[j]++;

else

count[i]++;

for (i = 0; i < size; i++)

b[count[i]] = rows[i];

for (i = 0; i < size; i++)

rows[i] = b[i];

}

void scramble(Row\* rows, const int size)

{

int i, j, z;

srand((unsigned int)time(0));

for (z = 0; z < size; z++)

{

i = randomAB(0, size - 1);

j = randomAB(0, size - 1);

swapRows(rows, i, j);

}

}

void reverse(Row\* rows, const int size)

{

int i, j;

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

swapRows(rows, i, j);

}

void getRow(FILE\* stream, char\* str, const int size)

{

int cnt = 0, ch;

while ((ch = getc(stream)) != '\n' && cnt < size - 1)

str[cnt++] = ch;

str[cnt] = '\0';

}

void swapRows(Row\* rows, const int a, const int b)

{

Row tmp;

tmp = rows[a];

rows[a] = rows[b];

rows[b] = tmp;

}

int comparator(const Key k1, const Key k2)

{

if (k1.a > k2.a)

return 1;

if (k1.a < k2.a)

return -1;

if (k1.b > k2.b)

return 1;

if (k1.b < k2.b)

return -1;

return 0;

}

int randomAB(const int a, const int b)

{

return a + rand() % (b - a + 1);

}

int isSorted(const Row\* rows, const int size)

{

int i;

for (i = 0; i < size - 1; i++)

if (comparator(rows[i].key, rows[i + 1].key) > 0)

return 0;

return 1;

}

int main(void)

{

const int N = 50;

int i, cnt, action;

char ch;

Row data[N];

Key key;

FILE\* file = fopen("input.txt", "r");

if (file == NULL)

{

printf("Can not open file\n");

return 0;

}

i = 0;

while (i < N && fscanf(file, "%hd %hd", &data[i].key.a, &data[i].key.b) == 2)

{

fscanf(file, "%c", &ch);

getRow(file, data[i].val, sizeof(data[i].val));

i++;

}

fclose(file);

cnt = i;

do

{

printf("Menu\n");

printf("1) Print\n");

printf("2) Binary search\n");

printf("3) Sort\n");

printf("4) Mix\n");

printf("5) Reverse\n");

printf("6) Quit\n");

printf("Your choice:\n");

scanf("%d", &action);

switch (action)

{

case 1:

{

printTable(data, cnt);

break;

}

case 2:

{

if (!isSorted(data, cnt))

printf("Table is not sorted\n");

else

{

printf("Enter key: ");

scanf("%hd %hd", &key.a, &key.b);

i = binSearch(data, cnt, key);

if (i > -1)

printf("String find: %s\n", data[i].val);

else

printf("String not find\n");

}

break;

}

case 3:

{

sort(data, cnt);

break;

}

case 4:

{

scramble(data, cnt);

break;

}

case 5:

{

reverse(data, cnt);

break;

}

case 6: break;

default:

{

printf("Wrong choice\n");

break;

}

}

} while (action != 6);

return 0;

}

**Протокол**

anri@DESKTOP-P2DRD4H:~/kp9$ cat > input.txt

100 10 Music to hear, why hear'st thou music sadly?

100 11 Sweets with sweets war not, joy delights in joy.

101 12 Why lovest thou that which thou receivest not gladly,

102 12 Or else receivest with pleasure thine annoy?

103 13 If the true concord of well-tuned sounds,

103 14 By unions married, do offend thine ear,

104 15 They do but sweetly chide thee, who confounds

105 15 In singleness the parts that thou shouldst bear.

106 16 Mark how one string, sweet husband to another,

106 17 Strikes each in each by mutual ordering,

107 18 Resembling sire and child and happy mother

108 18 Who all in one, one pleasing note do sing:

109 19 Whose speechless song, being many, seeming one,

109 20 Sings this to thee: "thou single wilt prove none."

anri@DESKTOP-P2DRD4H:~/kp9$ gcc kp9.c -o kp9.exe

anri@DESKTOP-P2DRD4H:~/kp9$ ./kp9.exe

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

1

+---------+-----------------------------------------------------------------------------+

| Key | string |

+---------+-----------------------------------------------------------------------------+

| 100 10| Music to hear, why hear'st thou music sadly?

| 100 11| Sweets with sweets war not, joy delights in joy.

| 101 12| Why lovest thou that which thou receivest not gladly,

| 102 12| Or else receivest with pleasure thine annoy?

| 103 13| If the true concord of well-tuned sounds,

| 103 14| By unions married, do offend thine ear,

| 104 15| They do but sweetly chide thee, who confounds

| 105 15| In singleness the parts that thou shouldst bear.

| 106 16| Mark how one string, sweet husband to another,

| 106 17| Strikes each in each by mutual ordering,

| 107 18| Resembling sire and child and happy mother

| 108 18| Who all in one, one pleasing note do sing:

| 109 19| Whose speechless song, being many, seeming one,

| 109 20| Sings this to thee: "thou single wilt prove none."

+---------+-----------------------------------------------------------------------------+

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

3

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

1

+---------+-----------------------------------------------------------------------------+

| Key | string |

+---------+-----------------------------------------------------------------------------+

| 100 10| Music to hear, why hear'st thou music sadly?

| 100 11| Sweets with sweets war not, joy delights in joy.

| 101 12| Why lovest thou that which thou receivest not gladly,

| 102 12| Or else receivest with pleasure thine annoy?

| 103 13| If the true concord of well-tuned sounds,

| 103 14| By unions married, do offend thine ear,

| 104 15| They do but sweetly chide thee, who confounds

| 105 15| In singleness the parts that thou shouldst bear.

| 106 16| Mark how one string, sweet husband to another,

| 106 17| Strikes each in each by mutual ordering,

| 107 18| Resembling sire and child and happy mother

| 108 18| Who all in one, one pleasing note do sing:

| 109 19| Whose speechless song, being many, seeming one,

| 109 20| Sings this to thee: "thou single wilt prove none."

+---------+-----------------------------------------------------------------------------+

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

2

Enter key: 109 19

String find: Whose speechless song, being many, seeming one,

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

4

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

1

+---------+-----------------------------------------------------------------------------+

| Key | string |

+---------+-----------------------------------------------------------------------------+

| 103 14| By unions married, do offend thine ear,

| 106 16| Mark how one string, sweet husband to another,

| 101 12| Why lovest thou that which thou receivest not gladly,

| 109 20| Sings this to thee: "thou single wilt prove none."

| 107 18| Resembling sire and child and happy mother

| 104 15| They do but sweetly chide thee, who confounds

| 102 12| Or else receivest with pleasure thine annoy?

| 106 17| Strikes each in each by mutual ordering,

| 103 13| If the true concord of well-tuned sounds,

| 100 11| Sweets with sweets war not, joy delights in joy.

| 105 15| In singleness the parts that thou shouldst bear.

| 108 18| Who all in one, one pleasing note do sing:

| 109 19| Whose speechless song, being many, seeming one,

| 100 10| Music to hear, why hear'st thou music sadly?

+---------+-----------------------------------------------------------------------------+

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

2

Table is not sorted

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

3

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

1

+---------+-----------------------------------------------------------------------------+

| Key | string |

+---------+-----------------------------------------------------------------------------+

| 100 10| Music to hear, why hear'st thou music sadly?

| 100 11| Sweets with sweets war not, joy delights in joy.

| 101 12| Why lovest thou that which thou receivest not gladly,

| 102 12| Or else receivest with pleasure thine annoy?

| 103 13| If the true concord of well-tuned sounds,

| 103 14| By unions married, do offend thine ear,

| 104 15| They do but sweetly chide thee, who confounds

| 105 15| In singleness the parts that thou shouldst bear.

| 106 16| Mark how one string, sweet husband to another,

| 106 17| Strikes each in each by mutual ordering,

| 107 18| Resembling sire and child and happy mother

| 108 18| Who all in one, one pleasing note do sing:

| 109 19| Whose speechless song, being many, seeming one,

| 109 20| Sings this to thee: "thou single wilt prove none."

+---------+-----------------------------------------------------------------------------+

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

5

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

1

+---------+-----------------------------------------------------------------------------+

| Key | string |

+---------+-----------------------------------------------------------------------------+

| 109 20| Sings this to thee: "thou single wilt prove none."

| 109 19| Whose speechless song, being many, seeming one,

| 108 18| Who all in one, one pleasing note do sing:

| 107 18| Resembling sire and child and happy mother

| 106 17| Strikes each in each by mutual ordering,

| 106 16| Mark how one string, sweet husband to another,

| 105 15| In singleness the parts that thou shouldst bear.

| 104 15| They do but sweetly chide thee, who confounds

| 103 14| By unions married, do offend thine ear,

| 103 13| If the true concord of well-tuned sounds,

| 102 12| Or else receivest with pleasure thine annoy?

| 101 12| Why lovest thou that which thou receivest not gladly,

| 100 11| Sweets with sweets war not, joy delights in joy.

| 100 10| Music to hear, why hear'st thou music sadly?

+---------+-----------------------------------------------------------------------------+

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

3

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

1

+---------+-----------------------------------------------------------------------------+

| Key | string |

+---------+-----------------------------------------------------------------------------+

| 100 10| Music to hear, why hear'st thou music sadly?

| 100 11| Sweets with sweets war not, joy delights in joy.

| 101 12| Why lovest thou that which thou receivest not gladly,

| 102 12| Or else receivest with pleasure thine annoy?

| 103 13| If the true concord of well-tuned sounds,

| 103 14| By unions married, do offend thine ear,

| 104 15| They do but sweetly chide thee, who confounds

| 105 15| In singleness the parts that thou shouldst bear.

| 106 16| Mark how one string, sweet husband to another,

| 106 17| Strikes each in each by mutual ordering,

| 107 18| Resembling sire and child and happy mother

| 108 18| Who all in one, one pleasing note do sing:

| 109 19| Whose speechless song, being many, seeming one,

| 109 20| Sings this to thee: "thou single wilt prove none."

+---------+-----------------------------------------------------------------------------+

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

2

Enter key: 106 17

String find: Strikes each in each by mutual ordering,

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

2

Enter key: 100 10

String find: Music to hear, why hear'st thou music sadly?

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

2

Enter key: 104 15

String find: They do but sweetly chide thee, who confounds

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

6

Menu

1) Print

2) Binary search

3) Sort

4) Mix

5) Reverse

6) Quit

Your choice:

2

Enter key: 123 10

String not find

anri@DESKTOP-P2DRD4H:~/kp9$

**Заключение**

В процессе выполнения данного задания я познакомился с алгоритмами сортировки и бинарного поиска, научился с ними работать.