Отчёт по заданию №9 курсового проекта

Черыгова Елизавета
Группа 8О-104Б

Оглавление

Цель работы

Составить программу на Си для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице.

Входные данные

На первой строке находится число M, указывающее количество записей таблице. На следующих M строках находятся пары ключ-значение, разделенные знаком табуляции. Типы ключа и значения зависят от варианта задания. Далее до конца файла находятся ключи, которые нужно искать в таблице.

Выходные данные

Таблица, состоящая из тех же строк, что и входная, но расположенных в отстортированном порядке. Для каждого ключа, который нужно было найти в таблице вывести соответствующие значения, разделенные знаком табуляции или "Not found", если такого ключа в таблице нет.

Задание:

Сортировка: метод пузырька.

Структура таблицы: строковые ключи массива, 5 байт, хранение отдельно, элементов таблицы 8-12.

Алгоритм решения задачи

Сортировка пузырьком

Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции как пузырёк в воде, отсюда и название алгоритма).

Сложность алгоритма: $O(n^2)$.

Код программы

```
/*      Сортировка:                     3. Метод пузырька.
        Структура таблицы:  3. Ключ-строковы, 5 байт, хранение отдельно, эл-тов таблицы
8-12.
 */

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include "function9.h"

int main(){
        FILE* f;
        char fname[50];
        char fkey[6];
        t_key key[12];
        t_val value[12];
        int size=0;
        // Вводим имя файла.
        printf("Enter filename: ");
   scanf("%s",fname);
   f=fopen(fname,"r");
   if(f==NULL){
      printf("File doesn't exist");
      return 0;
   }
   // Печатаем содержимое файла.
   printf("\nDefault table:\n");
   while(fscanf(f,"%s %s",key[size].key,value[size].val)!=EOF){
      printf("%s %s\n",key[size].key,value[size].val);
      size++;
   }
   // Печатаем отсортированное содержимое файла.
   printf("\nSorting table:\n");
   bubble_sort(key ,value, size);
   int i=0;
   for(;i<size;i++){
        printf("%s %s\n",key[i].key,value[i].val);
   }
   // Вводим ключ, которому соответствует 2 значения.
   printf("\nEnter key to find\n");
   scanf("%s", fkey);
   int k=0;
   int felem=0;
   while(felem!=-1){
```

```
        felem=binary_search(key, fkey, size);
        if(felem!=-1){
                        printf("This element: \n%s\n", value[felem].val);
        // Первое значение выводится и свапается с последним
                strcpy(key[felem].key,key[size-1].key);
                strcpy(value[felem].val,value[size-1].val);
                bubble_sort(key ,value, size);
                size--;
                                // Размер уменьшаем.
        }
                                                // Каждый раз выводится первый эл-т, соотв. ключу.
Затем он удаляется и печатается след. и т.д.
        k++;
   }
   if(k==1)printf("NOT FOUND");


        return 0;
}
```

```
#ifndef FUNCTION7_H
#define FUNCTION7_H

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

typedef struct{
	char key[6]
}t_key;

typedef struct{
	char val[100];
}t_val;

/*******Объявление функций.*********/
void swapf(char* a, char* b);                                   // swap для ключей.
void swapv(char* a, char* b);                                   // swap для
значений.
void bubble_sort(t_key* skey, t_val* sval, int size);// Сортировка пузырьком массива ключей
skey размера size.
int binary_search(t_key* key, char* x, size_t size);  // Двоичный поиск в массиве key размера
size по ключу x.

/*********************************/

void swapf(char* a, char* b){
	char temp[6];
	strcpy(temp, a);
	strcpy(a, b);
	strcpy(b, temp);
	return;
}

void swapv(char* a, char* b){
	char temp[100];
	strcpy(temp, a);
	strcpy(a, b);
	strcpy(b, temp);
	return;
}

void bubble_sort(t_key* skey, t_val* sval, int size) {
	int i, j;
   for (j = 0; j < size-1; j++){
      for (i = 0; i < size - j - 1; i++){
```

```
            if (strcmp(skey[i].key,skey[i+1].key)>0){
                swapf(skey[i].key, skey[i+1].key);
                swapv(sval[i].val, sval[i+1].val);
            }
        }
    }
}


int binary_search(t_key* key, char* x, size_t size)
{
    size_t first = 0;
    size_t last = size;

    if (size == 0) {
        return -1;
    }
            else if (strcmp(key[0].key,x) > 0) {
        return -1;
    }
            else if (strcmp(key[size - 1].key, x) < 0) {
        return -1;
    }
    while (first < last) {
        size_t mid = first + (last - first) / 2;
        if (strcmp(x,key[mid].key)<=0)
            last = mid;
        else
            first = mid + 1;
    }
    if (strcmp(key[last].key,x) == 0){
        return last;
            }
            else {
        return -1;
    }
}


#endif
```

Вывод программы

Файл test.txt

```
ddddd Or_to_take_arms_against_a_sea_of_troubles,
ccccc The_slings_and_arrows_of_outrageous_fortune,
fffff No_more;_and_by_a_sleep_to_say_we_end
jjjjj
To_sleep:_perchance_to_dream:_ay,_there's_the_rub;
hhhhh That_flesh_is_heir_to,_'tis_a_consummation
iiiii Devoutly_to_be_wish'd._To_die,_to_sleep;
aaaaa To_be,_or_not_to_be:_that_is_the_question:
ggggg The_heart-ache_and_the_housand_natural_shocks
eeeee And_by_opposing_end_them?_To_die:_to_sleep;
aaaaa Whether_'tis_nobler_in_the_mind_to_suffer
```

Протокол

```
Enter filename: test.txt

Default table:
ddddd Or_to_take_arms_against_a_sea_of_troubles,
ccccc The_slings_and_arrows_of_outrageous_fortune,
fffff No_more;_and_by_a_sleep_to_say_we_end
jjjjj
To_sleep:_perchance_to_dream:_ay,_there's_the_rub;
hhhhh That_flesh_is_heir_to,_'tis_a_consummation
iiiii Devoutly_to_be_wish'd._To_die,_to_sleep;
aaaaa To_be,_or_not_to_be:_that_is_the_question:
ggggg The_heart-ache_and_the_housand_natural_shocks
eeeee And_by_opposing_end_them?_To_die:_to_sleep;
aaaaa Whether_'tis_nobler_in_the_mind_to_suffer

Sorting table:
aaaaa To_be,_or_not_to_be:_that_is_the_question:
aaaaa Whether_'tis_nobler_in_the_mind_to_suffer
ccccc The_slings_and_arrows_of_outrageous_fortune,
ddddd Or_to_take_arms_against_a_sea_of_troubles,
eeeee And_by_opposing_end_them?_To_die:_to_sleep;
fffff No_more;_and_by_a_sleep_to_say_we_end
ggggg The_heart-ache_and_the_housand_natural_shocks
hhhhh That_flesh_is_heir_to,_'tis_a_consummation
iiiii Devoutly_to_be_wish'd._To_die,_to_sleep;
jjjjj
To_sleep:_perchance_to_dream:_ay,_there's_the_rub;

Enter key to find
aaaaa
```

```
This element:
To_be,_or_not_to_be:_that_is_the_question:
This element:
Whether_'tis_nobler_in_the_mind_to_suffer
```

Заключение

Благодаря, этому заданию, я смогла поближе познакомиться и разобраться с представлением в Си программ для сортировки таблицы методом пузырька и двоичного поиска по ключу в таблице.