Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №16**

Дисциплина: «Основы теории алгоритмов и структуры данных»  
Тема: Методы внешней сортировки: метод прямого слияния, сбалансированного слияния, метод многофазной сортировки сортировка подсчётом

Вариант 14

Выполнил:

студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

доцент кафедры ИТАС

Полякова О. А.

Пермь, 2021

**Цель работы**

Получить практические навыки работы с методами внешних сортировок.

**Постановка задачи**

1. Сформировать массив чисел.

2. Отсортировать массив чисел методом естественного слияния;

3. Отсортировать массив чисел методом сбалансированного слияния;

4. Отсортировать массив чисел методом многофазного слияния.

**Анализ задачи**

1. Для решения задачи необходимо:

1.1. Сформировать массив чисел:

int \*m;

int r;

struct l{

int n,k;

l\* next;

};

void Create(){

cout << "Размерность массива: ";

cin >> r;

while (r<2||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Размерность массива должна быть более 1! Повторите ввод: ";

cin >> r;

}

cout << endl;

m = new int [r];

srand(time(0));

for (int i=0;i<r;i++){

m[i]=rand()%1000;

cout << m[i] << " ";

}

cout << endl;

}

1.2. Создать функцию, сортирующую массив методом естественного слияния:

void Est(){

l \*first, \*cur, \*nov;

int i=1;

first = new l;

first->n=0;

first->next=NULL;

cur=first;

while (i<r){

while (m[i]>=m[i-1]&&i<r) i++;

cur->k=i-1;

if (i<r){

nov = new l;

cur->next=nov;

cur=nov;

cur->n=i;

cur->next=NULL;

}

i++;

}

cur->k=r-1;

cur=first;

int pos,min;

while (first->next!=NULL){

cur=first;

while (cur->next!=NULL){

for (int e=cur->next->n,h=cur->next->k+1;e<h;e++){

for (int g=cur->n;g<h;g++){

if (m[e]<m[g]){

min=m[g];

m[g]=m[e];

m[e]=min;

}

}

}

for (int e=cur->n,h=cur->next->k+1;e<h;e++) cout << m[e] << " ";//!\_!

cout << endl;//!\_!

l \*del=cur->next;

cur->k=cur->next->k;

if (cur->next->next!=NULL) cur->next=cur->next->next;

else cur->next=NULL;

delete del;

if (cur->next!=NULL) cur=cur->next;

}

}

}

1.3. Создать функцию, сортирующую список методом Хоара:  
void Hoar(int l, int ri){

int sl=l,sr=ri,o=(ri-l)/2+l;

string h;

if (l<ri){

while (l!=ri&&ri-l>1){

while (m[l]<=m[o]&&l<o) l++;

while (m[ri]>=m[o]&&ri>o) ri--;

h=m[l];m[l]=m[ri];m[ri]=h;

if ((ri==o||l==o)&&l!=ri) {

l=sl;ri=sr;

}

for (int i=0; i<r;i++) cout << m[i] << " ";

cout << endl;

}

if ((sr-sl)>1){

Hoar(sl,o);

Hoar(o,sr);

}

}

}

2. В ходе работы были использованы следующие типы данных:

2.1. Для хранения значений элементов массива и хранения нужных индексов элементов используется тип int:  
int \*m;

int r;

3. Для работы со связанным списком использовалась структура:

struct l{

int n,k;

l\* next;

};

4. Для операций ввода и вывода использовались следующие операторы:

4.1. Для ввода данных использовался оператор cin:

cin >> r;

4.2. Для вывода значений на экран используется оператор cout:

cout << m[i] << " ";

5. Для решения задач использовались следующие циклы:

5.1. Цикл while для проверки правильности ввода:

while (cin.fail()||r<2){

cout << "Размерность массива должна иметь значение не менее 2! Повторите ввод: ";

cin.ignore(30,'\n');

cin.clear();

cin >> r;

}

5.2. Для вывода данных используется цикл for:

for (int i=n;i<k+1;i++) cout << m[i] << " ";

**Полный код программы**

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <fstream>

using namespace std;

int \*m;

int r;

struct l{

int n,k;

l\* next;

};

void Create(){

cout << "Размерность массива: ";

cin >> r;

while (r<2||cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Размерность массива должна быть более 1! Повторите ввод: ";

cin >> r;

}

cout << endl;

m = new int [r];

srand(time(0));

for (int i=0;i<r;i++){

m[i]=rand()%1000;

cout << m[i] << " ";

}

cout << endl;

}

void Est(){

l \*first, \*cur, \*nov;

int i=1;

first = new l;

first->n=0;

first->next=NULL;

cur=first;

while (i<r){

while (m[i]>=m[i-1]&&i<r) i++;

cur->k=i-1;

if (i<r){

nov = new l;

cur->next=nov;

cur=nov;

cur->n=i;

cur->next=NULL;

}

i++;

}

cur->k=r-1;

cur=first;

int pos,min;

while (first->next!=NULL){

cur=first;

while (cur->next!=NULL){

for (int e=cur->next->n,h=cur->next->k+1;e<h;e++){

for (int g=cur->n;g<h;g++){

if (m[e]<m[g]){

min=m[g];

m[g]=m[e];

m[e]=min;

}

}

}

for (int e=cur->n,h=cur->next->k+1;e<h;e++) cout << m[e] << " ";//!\_!

cout << endl;//!\_!

l \*del=cur->next;

cur->k=cur->next->k;

if (cur->next->next!=NULL) cur->next=cur->next->next;

else cur->next=NULL;

delete del;

if (cur->next!=NULL) cur=cur->next;

}

}

}

void Sbalanc(int n, int k){

int min;

if (k-n>1){

Sbalanc(n,(k+n)/2);

Sbalanc((k+n)/2+1,k);

}

for (int i=(k+n)/2+1;i<k+1;i++){

min=m[i];

for (int e=n;e<k+1;e++){

if (m[i]<m[e]){

min=m[e];

m[e]=m[i];

m[i]=min;

}

}

}

for (int i=n;i<k+1;i++) cout << m[i] << " ";

cout << endl;

}

void MFaz(){

const int dlina=100;

int ser[3];

int fi=1, bo=1, na, o;

cout << "Размерность массива (r>199): ";

cin >> r;

while (cin.fail()||r<200){

cin.ignore(30,'\n');

cin.clear();

cout << "Значение длины массива должно быть не менее 200! Повторите ввод: ";

cin >> r;

}

int \*z= new int [r];

for (int i=0;i<r;i++) z[i]=rand()%10000+1;

while (r>(100\*(fi+bo))){

na=fi;

fi+=bo;

bo=na;

}

ser[0]=fi;

ser[1]=bo;

ser[2]=0;

ofstream o1 ("1.txt");

ofstream o2 ("2.txt");

for (int i=0; i<ser[0];i++){

int e=0;

o1 << z[i\*100+e] << ' ';

e++;

while ((i\*100+e)%100){

o1 << z[i\*100+e] << ' ';

e++;

}

o1 << -1 << endl;

}

o1.close();

for (int i=0; i<ser[1];i++){

int e=0;

if (i\*100+fi\*100+e<r) o2 << z[i\*100+fi\*100+e] << ' ';

e++;

while ((i\*100+fi\*100+e)%100&&(i\*100+fi\*100+e)<r){

o2 << z[i\*100+fi\*100+e] << ' ';

e++;

}

o2 << -1 << endl;

}

o2.close();

o=2;

ifstream i1;

ifstream i2;

ofstream out;

int leng=200;

fi=1;bo=1;na=1;

while (ser[0]+ser[1]+ser[2]>1){

cout << ser[0] << " " << ser[1] << " " << ser[2] << endl;//!\_!

cout << leng << endl;//!\_!

int \*kesh = new int [leng];

if (o==2){

out.open("3.txt");

i1.open("1.txt");

i2.open("2.txt");

o--;

cout << "Вывод в 3-й файл. " << endl;

}

else if (o==1){

out.open("2.txt");

i1.open("1.txt");

i2.open("3.txt");

o--;

cout << "Вывод во 2-й файл. " << endl;

}

else if (o==0){

out.open("1.txt");

i1.open("2.txt");

i2.open("3.txt");

o=2;

cout << "Вывод в 1-й файл. " << endl;

}

int elm=0, mins=100;

if (ser[0]<mins&&ser[0]) mins=ser[0];

if (ser[1]<mins&&ser[1]) mins=ser[1];

if (ser[2]<mins&&ser[2]) mins=ser[2];

for (int k=0;k<mins;k++){

int v=0;

for (int p=0;p<leng;p++) kesh[p]=0;

while (elm!=-1){

i1 >> elm;

if (elm!=-1) kesh[v++]=elm;

}

elm=0;

while (elm!=-1){

i2 >> elm;

if (elm!=-1) kesh[v++]=elm;

}

elm=0;

for (int i=0;i<leng;i++){

int min=kesh[i],pos=i;

for (int e=i;e<leng;e++){

if (kesh[e]<min){

pos=e;

min=kesh[e];

}

}

if (kesh[i]>min){

kesh[pos]=kesh[i];

kesh[i]=min;

}

}

for (int i=0;i<leng;i++)

if (kesh[i]){

out << kesh[i] << ' ';

}

out << -1 << endl;

if (o==1) {ser[0]--;ser[1]--;ser[2]++;}

else if (o==0) {ser[0]--;ser[2]--;ser[1]++;}

else if (o==2) {ser[1]--;ser[2]--;ser[0]++;}

}

int \*rez = new int [r];

int elein=0;

ofstream rezer;

for (int u=0;u<r;u++) rez[u]=0;

while (!i1.eof()){

i1 >> rez[elein++];

}

elein=0;

i1.close();

if (o==1||o==0) rezer.open("1.txt");

else if (o==2) rezer.open("2.txt");

while (rez[elein]!=0){

while (rez[elein]!=-1){

rezer << rez[elein] << ' ';

elein++;

}

rezer << -1 << endl;

elein++;

}

rezer.close();

for (int u=0;u<r;u++) rez[u]=0;

elein=0;

while (!i2.eof()){

i2 >> rez[elein++];

}

elein=0;

i2.close();

if (o==0||o==2) rezer.open("3.txt");

else if (o==1) rezer.open("2.txt");

while (rez[elein]!=0){

while (rez[elein]!=-1){

rezer << rez[elein] << ' ';

elein++;

}

rezer << -1 << endl;

elein++;

}

rezer.close();

delete[] rez;

system("pause");

out.close();

na=fi;

fi+=bo;

bo=na;

leng=(fi+bo)\*100;

delete[] kesh;

}

}

void Set(){

setlocale (LC\_ALL,"");

const int fon = system("Color F0");

}

int main(){

Set();

Create();

Est();

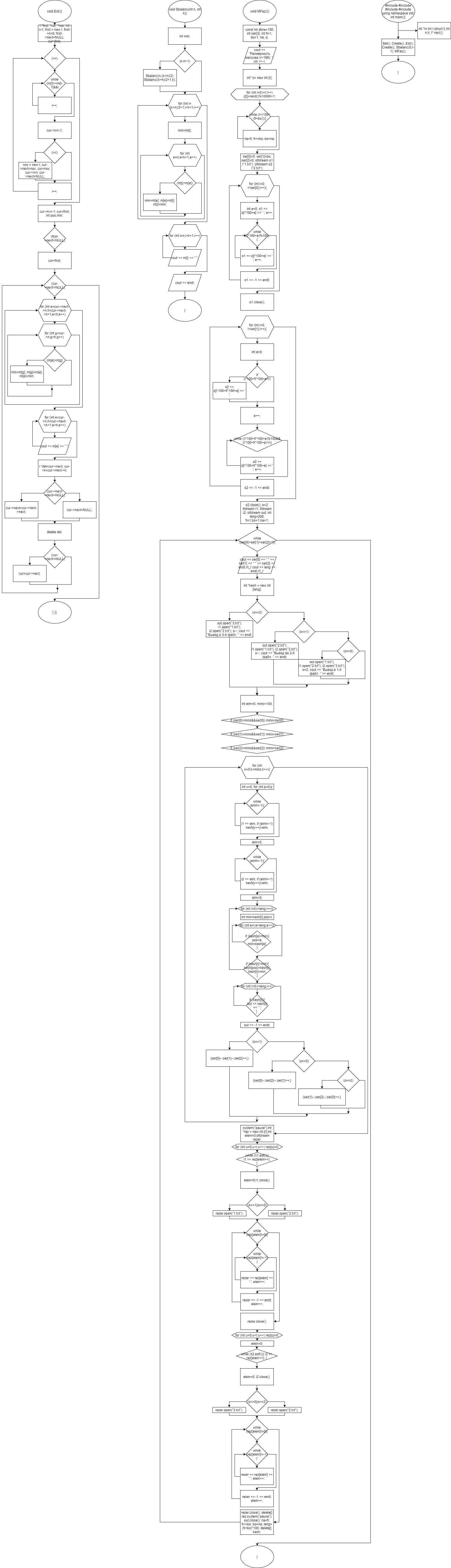
Create();

Sbalanc(0,r-1);

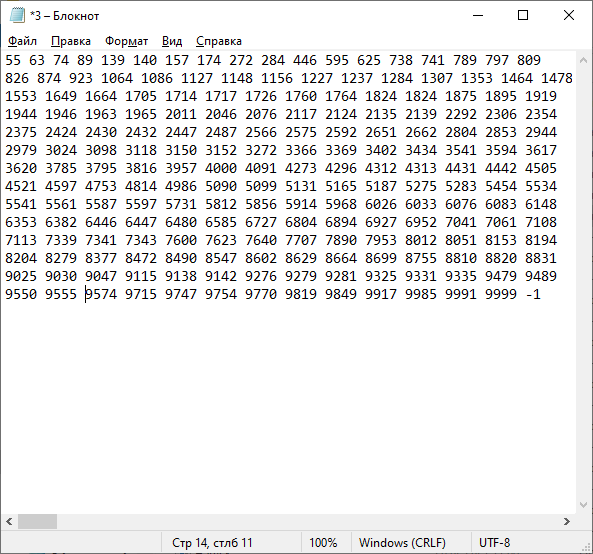
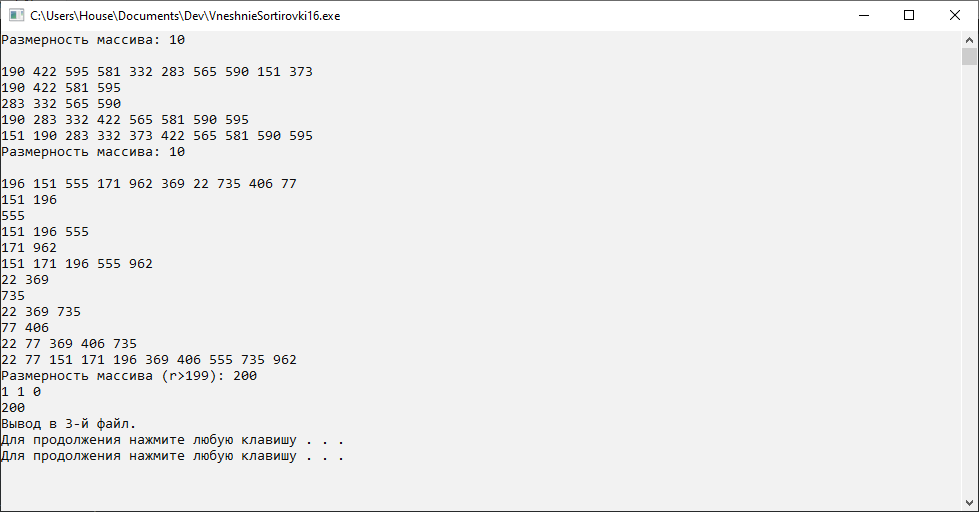
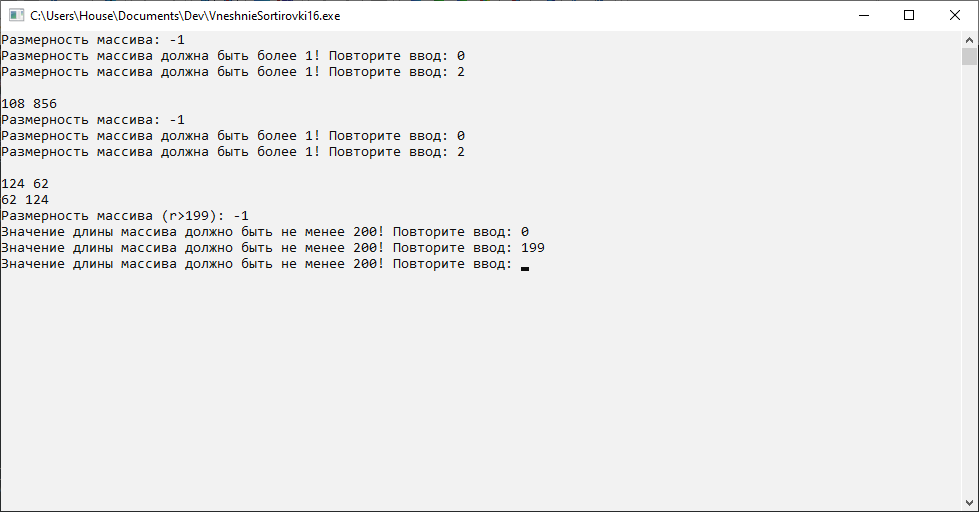
MFaz();

}

**Блок-схема**



**Результат работы программы**

****