**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра информационных систем**

**ОТЧЕТ**

**по лабораторной работе № 2**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: Реализация алгоритма сортировки Timsort**

Студент гр. 9375 Гоголев И.Л.

Преподаватель Лавренкова К.

Санкт-Петербург

2020

**Цель работы:**

1. Реализовать алгоритм сортировки Timsort.

**Текст программы:**

#include <iostream>

#include <windows.h>

#include <conio.h>

using namespace std;

//R - размер массива

//massive - упорядоченный подмассив во входном массиве

//minmassive - минимальный размер подмассива

int getMinmassive(int R) //получение минимального размера подмассива

{

int r = 0; // если среди сдвинутых битов будет хотя бы один ненулевой, станет 1

while (R >= 64) {

r |= R & 1;

R>>= 1;

}

return R + r;

}

void insertionSort(int Array[], int L, int R) //сортировка вставкой

{

for (int i = L + 1; i <= R; i++)

{

int temp = Array[i];

int j = i - 1;

while (j >= L && Array[j] > temp)

{

Array[j + 1] = Array[j];

j--;

}

Array[j + 1] = temp;

}

}

void merge(int Array[], int l, int m, int r) //сортировка подмассивов

{

//разбиваем подмассив на правую и левую части

int len1 = m - l + 1; //длинна левой части

int len2 = r - m; //длинна правой части

int\* left = new int[len1];

int\* right = new int[len2];

for (int i = 0; i < len1; i++)

left[i] = Array[l + i];

for (int i = 0; i < len2; i++)

right[i] = Array[m + 1 + i];

int i = 0;

int j = 0;

int k = l;

//после сравнения объединяются оба массива

while (i < len1 && j < len2)

{

if (left[i] <= right[j])

{

Array[k] = left[i];

i++;

}

else

{

Array[k] = right[j];

j++;

}

k++;

}

while (i < len1) { //Если остались элементы в левом массиве, копируем их

Array[k] = left[i];

k++;

i++;

}

while (j < len2) { //Если остались элементы в правом массиве, копируем их

Array[k] = right[j];

k++;

j++;

}

delete[] left;

delete[] right;

}

void timSort(int Array[], int R)

{

int minRun = getMinmassive(R);

for (int i = 0; i < R; i += minRun) { //сортировка подмассивов размера MAS

int min;

if ((i + (minRun - 1)) >= (R - 1)) {

min = R- 1;

}

else {

min = i + (minRun - 1);

}

insertionSort(Array, i, min);

}

for (int size = minRun; size < R; size = 2 \* size)

{

// выбираем точку начала левого подмассива, объединяем Array[left..left+size-1] и Array[left+size, left+2\*size-1]

// увеличиваем левый подмассив на 2\*size после каждого слияния

for (int left = 0; left < R; left += 2 \* size)

{

// находим конечную точку левого подмассива. mid+1 - начальная точка правого подмассива

int min;

if (((left + 2) \* (size - 1)) >= (R - 1)) {

min = R- 1;

}

else {

min = (left + 2) \* (size - 1);

}

int mid = left + size - 1;

int right = min;

merge(Array, left, mid, right); // сливаем подмассивы Array[left.....mid] и Array[mid+1....right]

}

}

}

void inputrandom() {

cout << "Введите размер массива: ";

int arrSize;

cin >> arrSize;

int\* Array = new int[arrSize];

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

Array[i] = rand() % 10;

}

cout << "Начальный массив: ";

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

cout << Array[i] << ' ';

}

timSort(Array, arrSize);

cout << endl;

cout << "Преобразованный массив: ";

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

cout << Array[i] << ' ';

}

\_getch();

delete[]Array;

}

void inputconsole() {

cout << "Введите размер массива: ";

int arrSize;

cin >> arrSize;

int\* Array = new int[arrSize];

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

cout << "Array[" << i << "] = ";

cin >> Array[i];

}

cout << "Начальный массив: ";

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

cout << Array[i] << ' ';

}

timSort(Array, arrSize);

cout << endl;

cout << "Преобразованный массив: ";

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

cout << Array[i] << ' ';

}

\_getch();

delete[]Array;

}

void menu() {

system("cls");

cout << " \_\_\_\_\_\_\_\_\_\_\_ " << endl;

cout << " | |" << endl;

cout << " | Tim$ort |" << endl;

cout << " |\_\_\_\_\_\_\_\_\_\_\_|" << endl;

cout << endl;

cout << " ArrowUp | Рандомный массив |" << endl;

cout << "ArrowDown | Собственный массив |" << endl;

cout << " Esc | Выход |" << endl;

cout << endl;

}

int main()

{

setlocale(LC\_ALL, "rus");

int answer;

do {

menu();

answer = \_getch();

switch (answer) {

case 72:

inputrandom();

break;

case 80:

inputconsole();

break;

case 27:

break;

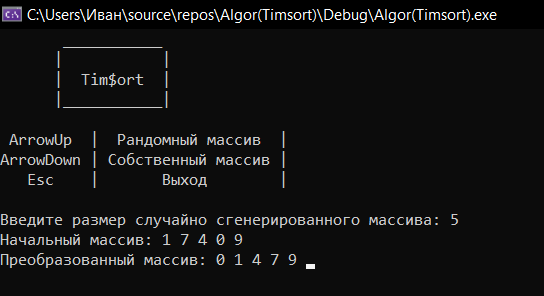
}

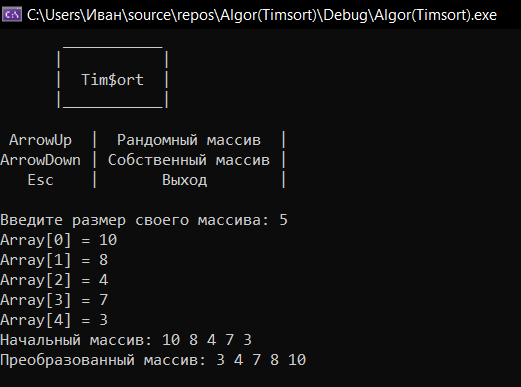
} while (answer != 27);

return 0;

}

**Пример запуска программы:**

****

****