# Учреждение образования

# «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Отчёт по лабораторной работе №16

Вариант 2

Выполнил:

студент 1 курса 2 группы Самцевич Алексей

Минск 2020

Задание №1. Ввести массив **А**, в массив **В** перенести все элементы массива **А**, имеющие четный индекс, справа от которых расположены элементы с нечетным значением. Массив **В** отсортировать по убыванию, используя алгоритмы сортировок: сортировка выбором, сортировка Хоара.

Код:

//

// main.cpp

// laba\_16.1

//

// Created by Alex Samtsevich on 08.06.2020.

// Copyright © 2020 Alex Samtsevich. All rights reserved.

//

#include <iostream>

#include <ctime>

using namespace std;

void selectSort(int\* A, int size);

int\* sortHoar(int\* A, int sm, int em);

void print(int\* mas, int n) {

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

if (i<7 || i>n - 7)

cout << mas[i] << " ";

else

cout << ".";

cout << endl;

}

int main() {

srand(time(0));

int n = 13;

int\* a = new int[n];

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

a[i] = rand() % 100;

int m = 0; // размер второго массив

for (int i = 0; i < n - 1; i += 2)

if (a[i + 1] % 2 != 0)

m++;//кол-во элементов в б

int\* b = new int[m];

int j = 0;

for (int i = 0; i < n - 1; i += 2)

if (a[i + 1] % 2 != 0)

b[j++] = a[i];

cout << "a = ";

print(a, n);

cout << "b = ";

print(b, m);

clock\_t begin = clock();

selectSort(b, m);

clock\_t end = clock();

cout << "sort b = ";

print(b, m);

cout << endl <<"Count: "<< m << " Time: " << ((double)(end-begin)/CLOCKS\_PER\_SEC)/1000 << endl;

delete[]b;

b = new int[m];

j = 0;

for (int i = 0; i < n - 1; i += 2)

if (a[i + 1] % 2 != 0)

b[j++] = a[i];

begin = clock();

sortHoar(b, 0, m-1);

end = clock();

cout << endl << "Count: " << m << " Time: " << ((double)(end - begin) / CLOCKS\_PER\_SEC) / 1000 << endl;

delete[]a;

delete[]b;

for (n = 2; n <= 100000; n \*= 1.5) {

int\* a = new int[n];

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

a[i] = rand() % 100;

clock\_t begin = clock();

selectSort(a, n);

clock\_t end = clock();

cout << n << " " << ((double)(end - begin) / CLOCKS\_PER\_SEC);

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

a[i] = rand() % 100;

begin = clock();

sortHoar(a, 0, n-1);

end = clock();

cout << " " << ((double)(end - begin) / CLOCKS\_PER\_SEC) << endl;

delete[]a;

}

}

void selectSort(int\* A, int size) // сортировка выбором

{

for (int begin = 0; begin < size - 1; begin++) {

int ind = begin;

for (int j = begin+1; j < size; j++)

if (A[ind] < A[j])

ind = j;

int t = A[ind];

A[ind] = A[begin];

A[begin] = t;

}

}

int getHoarBorder(int\* A, int sm, int em) // сортировка ’оара

{

int i = sm - 1, j = em + 1;

int brd = A[sm];

int buf;

while (i < j)

{

while (A[--j] > brd);

while (A[++i] < brd);

if (i < j)

{

buf = A[j];

A[j] = A[i];

A[i] = buf;

};

}

return j;

}

int\* sortHoar(int\* A, int sm, int em)

{

if (sm < em)

{

int hb = getHoarBorder(A, sm, em);

sortHoar(A, sm, hb);

sortHoar(A, hb + 1, em);

}

return A;

};

Скриншот:



