//№ 1

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

namespace ConsoleApp3

{

class Timing

{

TimeSpan duration;

TimeSpan[] threads;

public Timing()

{

duration = new TimeSpan(0);

threads = new TimeSpan[Process.GetCurrentProcess().Threads.Count];

}

public void StartTime()

{

GC.Collect();

GC.WaitForPendingFinalizers();

for (int i = 0; i < threads.Length; i++)

threads[i] = Process.GetCurrentProcess().Threads[i].UserProcessorTime;

}

public void StopTime()

{

TimeSpan tmp;

for (int i = 0; i < threads.Length; i++)

{

tmp = Process.GetCurrentProcess().Threads[i].UserProcessorTime.Subtract(threads[i]);

if (tmp > TimeSpan.Zero)

duration = tmp;

}

}

public TimeSpan Result()

{

return duration;

}

}

class Program

{

//алгоритм сортировки методом включения

static public void SortInsertion(double[] arr)

{

double tmp;

int F = arr.Length;

for (int k = 1; k < F; k++)

{

tmp = arr[k];

int t = k - 1;

while (t >= 0 && tmp < arr[t]) arr[t + 1] = arr[t--];

arr[t + 1] = tmp;

}

}

//Сортировка Шелла (Shell sort)

static public double[] ShellSort(double[] a)

{

int k, t, inc;

double temp;

inc = 3;

while (inc > 0)

{

for (k = 0; k < a.Length; k++)

{

t = k;

temp = a[k];

while ((t >= inc) && (a[t - inc] > temp))

{

a[t] = a[t - inc];

t = t - inc;

}

a[t] = temp;

}

if (inc / 2 != 0)

inc = inc / 2;

else if (inc == 1)

inc = 0;

else

inc = 1;

}

return a;

}

static void Main(string[] args)

{

double[] ar1 = new double[10];

Random rnd = new Random();

for (int k = 0; k < ar1.Length; k++)

{

ar1[k] = rnd.NextDouble();

}

Timing objT = new Timing();

Stopwatch stpWatch = new Stopwatch();

objT.StartTime();

stpWatch.Start();

SortInsertion(ar1);

stpWatch.Stop();

objT.StopTime();

Console.WriteLine("Метод Включения; массив x10: ");

Console.WriteLine("StopWatch: " + stpWatch.Elapsed.ToString());

Console.WriteLine("Timing: " + objT.Result().ToString());

double[] ar2 = new double[1000];

Random rnd2 = new Random();

for (int k = 0; k < ar2.Length; k++)

{

ar2[k] = rnd2.NextDouble();

}

Timing objT2 = new Timing();

Stopwatch stpWatch2 = new Stopwatch();

objT2.StartTime();

stpWatch2.Start();

SortInsertion(ar2);

stpWatch2.Stop();

objT2.StopTime();

Console.WriteLine("Метод Включения; массив x1000: ");

Console.WriteLine("StopWatch: " + stpWatch2.Elapsed.ToString());

Console.WriteLine("Timing: " + objT2.Result().ToString());

double[] ar3 = new double[10];

Random rnd3 = new Random();

for (int k = 0; k < ar3.Length; k++)

{

ar3[k] = rnd3.NextDouble();

}

Timing objT3 = new Timing();

Stopwatch stpWatch3 = new Stopwatch();

objT3.StartTime();

stpWatch3.Start();

ShellSort(ar3);

stpWatch3.Stop();

objT3.StopTime();

Console.WriteLine("Метод Шелла; массив x10: ");

Console.WriteLine("StopWatch: " + stpWatch3.Elapsed.ToString());

Console.WriteLine("Timing: " + objT3.Result().ToString());

double[] ar4 = new double[1000];

Random rnd4 = new Random();

for (int k = 0; k < ar4.Length; k++)

{

ar4[k] = rnd4.NextDouble();

}

Timing objT4 = new Timing();

Stopwatch stpWatch4 = new Stopwatch();

objT4.StartTime();

stpWatch4.Start();

ShellSort(ar4);

stpWatch4.Stop();

objT4.StopTime();

Console.WriteLine("Метод Шелла; массив x1000: ");

Console.WriteLine("StopWatch: " + stpWatch4.Elapsed.ToString());

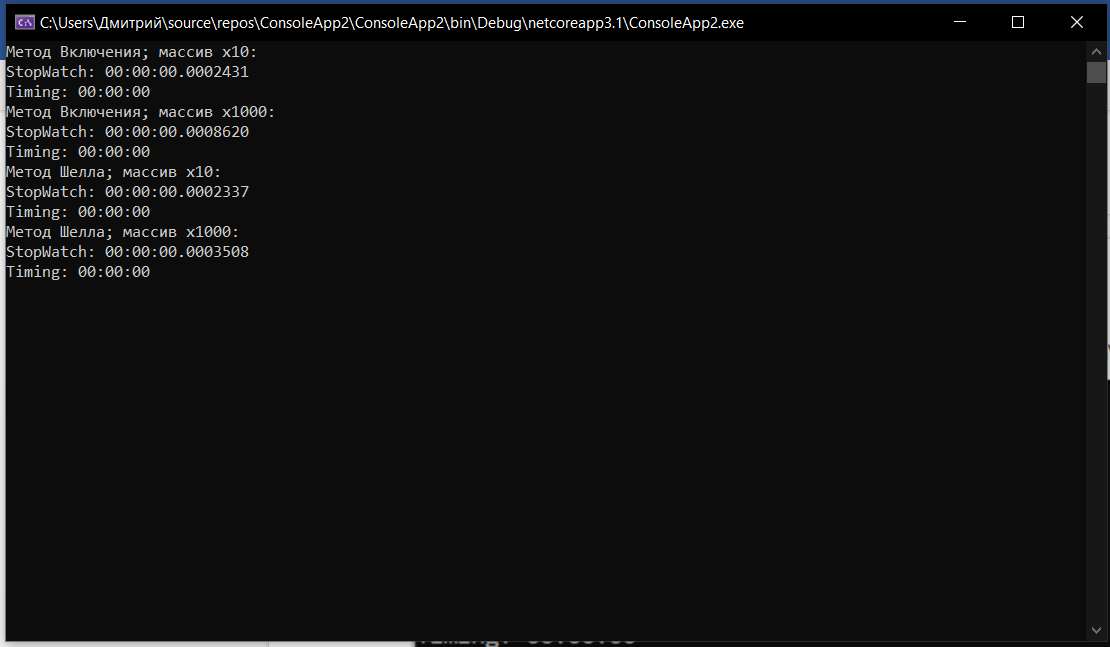
Console.WriteLine("Timing: " + objT4.Result().ToString());

Console.ReadLine();

}

}

}



//№ 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

namespace ConsoleApp3

{

class Timing

{

TimeSpan duration;

TimeSpan[] threads;

public Timing()

{

duration = new TimeSpan(0);

threads = new TimeSpan[Process.GetCurrentProcess().Threads.Count];

}

public void StartTime()

{

GC.Collect();

GC.WaitForPendingFinalizers();

for (int i = 0; i < threads.Length; i++)

threads[i] = Process.GetCurrentProcess().Threads[i].UserProcessorTime;

}

public void StopTime()

{

TimeSpan tmp;

for (int i = 0; i < threads.Length; i++)

{

tmp = Process.GetCurrentProcess().Threads[i].UserProcessorTime.Subtract(threads[i]);

if (tmp > TimeSpan.Zero)

duration = tmp;

}

}

public TimeSpan Result()

{

return duration;

}

}

class Program

{

// Простой поиск

static Double SearchSimple(double[] a, double x)

{

int P = a.Length;

int k = 0;

while (k < P && a[k] != x) k++;

if (k < P) return k;

else return -1;

}

// Бинарный поиск

static double SearchBinary(double[] a, double x)

{

int i, left = 0, right = a.Length - 1;

do

{

i = (left + right) / 2;

if (x > a[i]) left = i + 1;

else right = i - 1;

}

while ((a[i] != x) && (left <= right));

if (a[i] == x) return i;

else return -1;

}

static void Main(string[] args)

{

double[] ar1 = new double[10];

Random rnd = new Random();

for (int k = 0; k < ar1.Length; k++)

{

ar1[k] = rnd.NextDouble();

}

Timing objT = new Timing();

Stopwatch stpWatch = new Stopwatch();

objT.StartTime();

stpWatch.Start();

SearchSimple(ar1, 55.5);

stpWatch.Stop();

objT.StopTime();

Console.WriteLine("Простой поиск; массив x10: ");

Console.WriteLine("StopWatch: " + stpWatch.Elapsed.ToString());

Console.WriteLine("Timing: " + objT.Result().ToString());

double[] ar2 = new double[1000];

Random rnd2 = new Random();

for (int k = 0; k < ar2.Length; k++)

{

ar2[k] = rnd2.NextDouble();

}

Timing objT2 = new Timing();

Stopwatch stpWatch2 = new Stopwatch();

objT2.StartTime();

stpWatch2.Start();

SearchSimple(ar2, 55.5);

stpWatch2.Stop();

objT2.StopTime();

Console.WriteLine("Простой поиск; массив x1000: ");

Console.WriteLine("StopWatch: " + stpWatch2.Elapsed.ToString());

Console.WriteLine("Timing: " + objT2.Result().ToString());

double[] ar3 = new double[10];

Random rnd3 = new Random();

for (int k = 0; k < ar3.Length; k++)

{

ar3[k] = rnd3.NextDouble();

}

Timing objT3 = new Timing();

Stopwatch stpWatch3 = new Stopwatch();

objT3.StartTime();

stpWatch3.Start();

SearchBinary(ar3, 55.5);

stpWatch3.Stop();

objT3.StopTime();

Console.WriteLine("Бинарный поиск; массив x10: ");

Console.WriteLine("StopWatch: " + stpWatch3.Elapsed.ToString());

Console.WriteLine("Timing: " + objT3.Result().ToString());

double[] ar4 = new double[1000];

Random rnd4 = new Random();

for (int k = 0; k < ar4.Length; k++)

{

ar4[k] = rnd4.NextDouble();

}

Timing objT4 = new Timing();

Stopwatch stpWatch4 = new Stopwatch();

objT4.StartTime();

stpWatch4.Start();

SearchBinary(ar4, 55.5);

stpWatch4.Stop();

objT4.StopTime();

Console.WriteLine("Бинарный поиск; массив x1000: ");

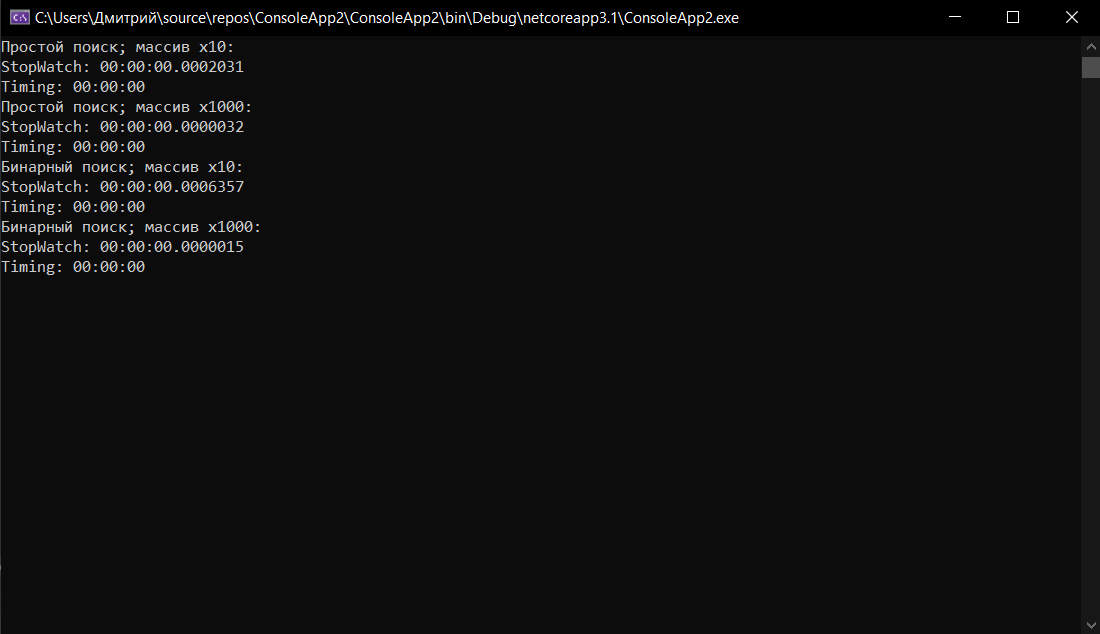
Console.WriteLine("StopWatch: " + stpWatch4.Elapsed.ToString());

Console.WriteLine("Timing: " + objT4.Result().ToString());

Console.ReadLine();

}

}

}

//№ 3

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

namespace ConsoleApp3

{

class Timing

{

TimeSpan duration;

TimeSpan[] threads;

public Timing()

{

duration = new TimeSpan(0);

threads = new TimeSpan[Process.GetCurrentProcess().Threads.Count];

}

public void StartTime()

{

GC.Collect();

GC.WaitForPendingFinalizers();

for (int i = 0; i < threads.Length; i++)

threads[i] = Process.GetCurrentProcess().Threads[i].UserProcessorTime;

}

public void StopTime()

{

TimeSpan tmp;

for (int i = 0; i < threads.Length; i++)

{

tmp = Process.GetCurrentProcess().Threads[i].UserProcessorTime.Subtract(threads[i]);

if (tmp > TimeSpan.Zero)

duration = tmp;

}

}

public TimeSpan Result()

{

return duration;

}

}

class Program

{

// Бинарный поиск

static double SearchBinary(double[] a, double x)

{

int i, left = 0, right = a.Length - 1;

do

{

i = (left + right) / 2;

if (x > a[i]) left = i + 1;

else right = i - 1;

}

while ((a[i] != x) && (left <= right));

if (a[i] == x) return i;

else return -1;

}

static void Main(string[] args)

{

double[] ar = new double[10];

Console.WriteLine("Введите массив");

for (int k = 0; k < ar.Length; k++)

{

ar[k] = double.Parse(Console.ReadLine());

}

Console.WriteLine("Введите число ");

int search = int.Parse(Console.ReadLine());

Timing objT = new Timing();

Stopwatch stpWatch = new Stopwatch();

objT.StartTime();

stpWatch.Start();

Console.WriteLine("Индекс искомого числа: " + SearchBinary(ar, search));

stpWatch.Stop();

objT.StopTime();

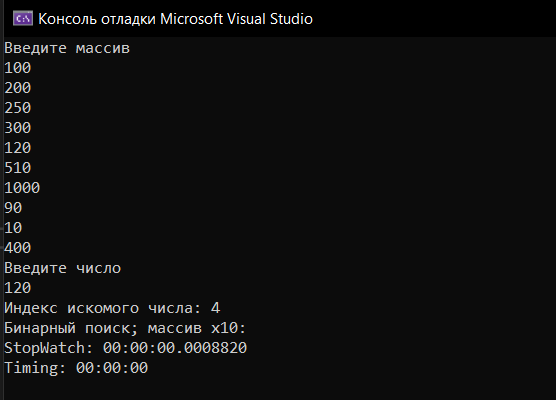
Console.WriteLine("Бинарный поиск; массив x10: ");

Console.WriteLine("StopWatch: " + stpWatch.Elapsed.ToString());

Console.WriteLine("Timing: " + objT.Result().ToString());

}

}

}

//№ 4

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

namespace ConsoleApp3

{

class Program

{

// Простой поиск

static Double SearchSimple(int[] a, double x)

{

int P = a.Length;

int k = 0;

while (k < P && a[k] != x) k++;

if (k < P) return k;

else return -1;

}

// Бинарный поиск

static double SearchBinary(int[] a, double x)

{

int i, left = 0, right = a.Length - 1, j = 0;

do

{

j++;

i = (left + right) / 2;

if (x > a[i]) left = i + 1;

else right = i - 1;

}

while ((a[i] != x) && (left <= right));

if (a[i] == x) return j;

else return -1;

}

// Поиск с барьером

static double SearchBarrier(int[] a, int x)

{

int P = a.Length;

Array.Resize<int>(ref a, ++P);

a[P - 1] = x;

int k = 0;

while (a[k] != x)

{

k++;

}

if (k < P - 1) return k;

else return -1;

}

static void Main(string[] args)

{

int[] array = new int[100];

Random rnd = new Random();

Console.WriteLine("Исходный массив: ");

for (int k = 0; k < array.Length; k++)

{

array[k] = k;

Console.Write(array[k] + " ");

}

Console.WriteLine(" ");

Random rnd2 = new Random();

int search = rnd2.Next(0, 100);

Console.WriteLine(search);

Console.WriteLine("Простой поиск: " + SearchSimple(array, search));

Console.WriteLine("Бинарный поиск: " + SearchBinary(array, search));

Console.WriteLine("Поиск с барьером: " + SearchBarrier(array, search));

}

}

}

