Выполнили студенты

215 группы

Овчарук Д.В.

Ботнарь А.С.

**Практическое задание №1**

Тема: **«Оценка сложности алгоритмов».**

**Цель занятия:** Сформировать и закрепить практические навыки оценки временной и емкостной сложности алгоритмов.

**Задание.**

В массиве из n целых чисел найти все пары элементов, сумма которых четна и сформировать новый массив из этих сумм.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using static System.Console;

namespace Practical\_task\_1

{

class Program

{

static int[] EvenSum(int[] array)

{

List<int> list\_1 = new List<int>();

//O(n) (n - длина массива)

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

{

//В худшем случае О(n), цикл не имеет квадратичной сложности, т.к. его выполнение зависит от индекса внешнего цикла.

for(int j = i + 1; j < array.Length; j++)

{

//Проверка на чётность: O(1) (выполняются операции за постоянное время)

if((array[i] + array[j]) % 2 == 0)

{

//Добавление в список: в среднем случае О(1) (иногда может произойти увеличение емкости списка, но не всегда)

list\_1.Add(array[i] + array[j]);

}

}

}

return list\_1.ToArray();

}

static void Main(string[] args)

{

try

{

Write("Enter n: ");

int n = Convert.ToInt32(ReadLine());

WriteLine();

int[] array = new int[n];

Random rnd = new Random();

//Временная сложность: O(n)

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

{

array[i] = rnd.Next(0, 255);

}

for (int i = 1; i < n; i++)

{

int j = i;

while (j > 0 && array[j] < array[j - 1])

{

int temp = array[j];

array[j] = array[j - 1];

array[j - 1] = temp;

j = j - 1;

}

}

//Временная сложность: O(n^2) (каждый элемент внешнего цикла вызывает внутренний цикл)

//Пространственная сложность: В худшем случае O(n^2)

int[] array\_2 = EvenSum(array);

foreach (var num in array)

{

Write($"{num};\t");

}

WriteLine($"\n");

foreach(var elem in array\_2)

{

Write($"{elem};\t");

}

}

catch (Exception ex)

{

WriteLine($"Исключение: {ex.Message}");

WriteLine($"Метод: {ex.TargetSite}");

WriteLine($"Трассировка стека: {ex.StackTrace}");

}

ReadKey();

}

}

}

**Вывод:** Мы сформировали и закрепили практические навыки оценки временной и емкостной сложности алгоритмов.

Проверила Балашова Ю.В.

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_