Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**дисциплины «Алгоритмизация»**

**Вариант\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Иващенко Олег Андреевич  2 курс, группа ИВТ-б-о-22-1,  09.03.02 «Информационные и вычислительные машины», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Доцент кафедры инфокоммуникации  Воронкин Роман Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема**: «Линейный поиск»

**Цель**: Создать программу для линейного поиска элемента в массиве.

Порядок выполнения работы:

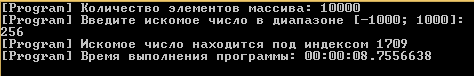


Рисунок 1 – Результат выполнения программы (значение найдено)

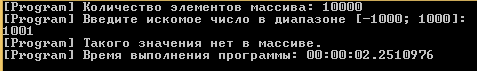


Рисунок 2 – Результат выполнения программы (значение не найдено)

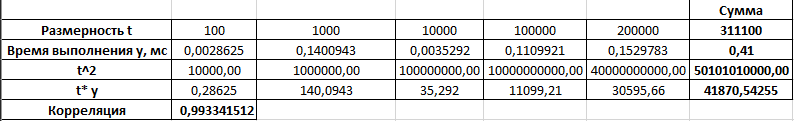


Рисунок 3 – Таблица с расчётом времени и корреляции для худших случаев (искомое значение последнее в массиве)

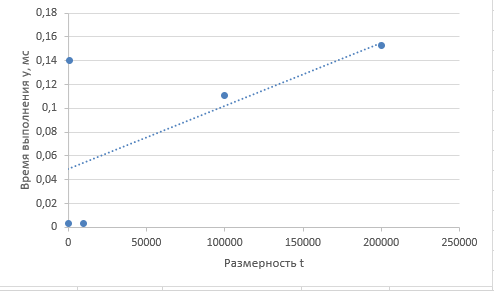


Рисунок 4 – График зависимости времени выполнения от количества элементов массива для худших случаев

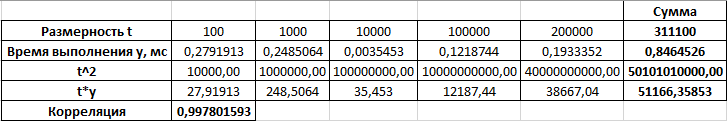


Рисунок 5 - Таблица с расчётом времени и корреляции для средних случаев (искомое значение находится в середине массива)

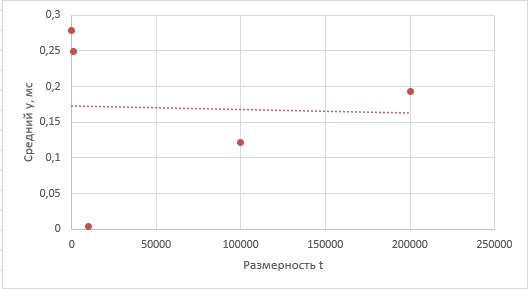


Рисунок 6 – График зависимости времени выполнения от количества элементов массива для средних случаев

Таблица 1 – Код программы

|  |
| --- |
| using System;  using System.Diagnostics;  class HelloWorld  {  static void Main()  {  var Timer = Stopwatch.StartNew();  int N = 10000;  Console.WriteLine($@"[Program] Количество элементов массива: {N}");  int[] Array = new int[N];  Random \_rnd = new Random();    Console.WriteLine("[Program] Введите искомое число в диапазоне [-1000; 1000]: ");  int SearchN = int.Parse(Console.ReadLine());  Timer.Start();  for (int i = 0; i < Array.Length; i++) Array[i] = \_rnd.Next(-1000, 1000); // Заполнение массива  int Result = SearchNumber(Array, SearchN); // Поиск элемента с указанным значением  if (Result == -1) Console.WriteLine("[Program] Такого значения нет в массиве."); // Случай, если элемента нет в массиве  else Console.WriteLine($"[Program] Искомое число находится под индексом {Result}"); // Случай, если элемент нашёлся  Console.WriteLine($"[Program] Время выполнения программы: {Timer.Elapsed}"); // Вывод  Timer.Stop();  Console.ReadKey();  }  static int SearchNumber(int[] NumberArray, int Number)  {  for (int i = 0; i < NumberArray.Length; i++) if (Number == NumberArray[i]) return i;  return -1;  }  } |