

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
Федерального государственного бюджетного образовательного учреждения
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

ЛАБОРАТОРНАЯ РАБОТА №3

По Структуры и алгоритмы обработки данных

Руководитель

Привезенцев Д.Г.

(фамилия, инициалы)

(подпись)

(дата)

Студент ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись)

(дата)

Муром 2023

Лабораторная работа №3

Тема: Исследование алгоритмов поиска

Ход работы:

1. Реализовать методы поиска данных в массиве:
 - a. Реализовать метод прямого поиска.
 - b. Реализовать метод бинарного поиска по массиву (сортировку массива производить любым алгоритмом).
 - c. Реализовать метод бинарного поиска с использованием бинарного дерева поиска
2. Реализовать управляющую программу(ы), включающую:
 - a. ввод исходных данных: из файла, с консоли, генерацией случайных чисел (способа ввода данных предусмотреть в программе путем ввода выбора пользователя, при этом размер массива тоже указывается во время выполнения программы);
 - b. ввод на экран исходных данных;
 - c. вывод на экран результата работы;
 - d. замер времени выполнения сортировки.
3. Выполнить исследование реализованных трех алгоритмах на разных коллекциях:
 - a. Выполнить замеры времени для коллекций размера 10, 100, 1 000, 10 000, 10 000, 1 000 000 000 элементов.
 - b. Заполнить таблицы 1 для каждого анализируемого алгоритма (всего три таблицы).
 - c. Сделать выводы по таблицам и графику.

					МИ ВлГУ 09.03.04			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Ермилов М.В.						
Провер.		Привезенцев Д.Г.						
Реценз.								
Н. Контр.								
Утверд.								
						Лит.	Лист	Листов
							2	5
						ПИН-121		

Таблица 1 - прямой поиск

Кол-во	10	100	1000	10 000	100 000	1 000 000	10 000 000	100 000 000	1 000 000 000
Операций	2	18	175	1748	17475	101292	1747496	1859455	145330
время	0.000208s	00.000000s	00.000000s	0.000011s	0.000110s	0.000621s	0.000443s	0.000816s	0.000576s

Таблица 2 - бинарный поиск

Кол-во	10	100	1000	10 000	100 000	1 000 000	10 000 000	100 000 000	1 000 000 000
Операций	2	6	9	11	16	18	22	20	20
время	0.000318s	00.000001s	00.000002s	0.000002s	0.000002s	0.000003s	0.000003s	0.000003s	0.000009s

Таблица 3 - поиск по бинарному дереву

Кол-во	10	100	1000	10 000	100 000	1 000 000	10 000 000	100 000 000	1 000 000 000
Операций	2	5	11	13	16	24	27	28	25
время	0.000203s	00.000000s	00.000000s	0.000001s	0.000005s	0.000003s	0.000002s	0.000002s	0.000003s

Код:

```
using System.Diagnostics;
```

```
int seed = 20;
int min = -1000000;
int max = min * -1;
```

```
Search(10);
Search(100);
Search(1000);
Search(10000);
Search(100000);
Search(1000000);
Search(10000000);
Search(100000000);
Search(1000000000);
```

```
void Search(int l)
{
    int count = 0;
    int search = 0;
    Console.WriteLine("Array Length = " + l);
    Stopwatch stopwatch = new Stopwatch();
    int[] arr = Rand(l, ref search);

    stopwatch.Start();
    Search1(arr, search, ref count);
    stopwatch.Stop();
    ConsoleWrite(count, stopwatch);

    int[] arr2 = Sort(arr);
    stopwatch.Restart();
    Search2(arr2, search, ref count);
    stopwatch.Stop();
    ConsoleWrite(count, stopwatch);

    BST arr3 = new BST(arr);
```

```

        stopwatch.Restart();
        Search3(arr3, search, ref count);
        stopwatch.Stop();
        ConsoleWrite(count, stopwatch);
    }

    int Search1(int[] arr, int digit, ref int count)
    {
        count = 0;
        for (int i = 0; i < arr.Length; i++)
        {
            count++;
            if (arr[i] == digit)
                return i;
        }
        return -1;
    }
    int Search2(int[] arr, int digit, ref int count)
    {
        count = 0;
        int L = arr.Length / 2;
        int l = L;
        while (L > 0)
        {
            L /= 2;
            count++;
            if (arr[l] > digit)
                l -= L;
            else if (arr[l] < digit)
                l += L;
            else
                return l;
        }
        return -1;
    }
    int Search3(BST a, int digit, ref int count)
    {
        count = 0;
        while (a != null)
        {
            count++;
            if (digit > a.Digit)
                a = a.Right;
            else if (digit < a.Digit)
                a = a.Left;
            else
                return a.Index;
        }
        return -1;
    }
    int[] Rand(int length, ref int search)
    {
        Random random = new Random(seed);
        int I = random.Next(0, length);
        int[] arr = new int[length];
        for (int i = 0; i < length; i++)
        {
            arr[i] = random.Next(min, max + 1);
            if (I == i)
                search = arr[i];
        }
        return arr;
    }
    int[] Sort(int[] arr)
    {
        int maxV = arr.Max();
    }

```

```

int minV = arr.Min();
int[] B = new int[maxV - minV + 1];
for (int i = 0; i < arr.Length; i++)
{
    B[arr[i] - minV]++;
}
int q = 0;
int[] newArr = new int[arr.Length];
for (int i = 0; i < (maxV - minV + 1); ++i)
{
    for (int j = 0; j < B[i]; ++j)
    {
        newArr[q++] = minV + i;
    }
}
return newArr;
}
void ConsoleWrite(int count, Stopwatch stopwatch)
{
    Console.WriteLine(
        $"{stopwatch.Elapsed.Seconds:00}" + "." +
        $"{stopwatch.Elapsed.Microseconds:000000}" + "sec;" +
        " Count = " + count
    );
}

```

```

class BST
{
    public BST Left = null;
    public BST Right = null;
    public int Index;
    public int Digit;
    public BST(int[] array)
    {
        for (int i = 0; i < array.Length; i++)
        {
            if (i == 0)
            {
                Digit = array[i];
                Index = i;
            }
            else
                Add(array[i], i);
        }
    }
    public BST(int d, int i)
    {
        Digit = d;
        Index = i;
    }
    public void Add(int d, int i)
    {
        if (d > Digit)
            if (Right != null)
                Right.Add(d, i);
            else
                Right = new BST(d, i);
        else if (d < Digit)
            if (Left != null)
                Left.Add(d, i);
            else
                Left = new BST(d, i);
    }
}

```