**Практическое занятие 3**

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

**Цель: научиться оценивать сложность рекурсивных алгоритмов.**

**Задание.**

**Вариант 4.**

**Разработать рекурсивный алгоритм поиска в отсортированном массиве и оценить его временную сложность в сравнении с итеративным поиском.**

class Program

{

static void Main(string[] args)

{

Console.Write("Введите число:");

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

try

{

int[] mass = new int[n];

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

{

Console.Write("Введите элемент массива:");

mass[i] = Convert.ToInt32(Console.ReadLine());

}

for(int i=0;i<mass.Length-1;i++)

{

for(int j=0;j<mass.Length-1-i;j++)

{

if(mass[j]>mass[j+1])

{

int temp = mass[j];

mass[j] = mass[j + 1];

mass[j+1]= temp;

}

}

}

foreach(var elem in mass)

{

Console.WriteLine($"Элемент массива:{elem}");

}

Console.Write("Введите искомое число:");

int unknow = Convert.ToInt32(Console.ReadLine());

if(-1== Rekurs(mass, unknow, 0, mass.Length - 1))

{

Console.WriteLine($"Искомое число не найдено");

}

else

{

Console.WriteLine($"Искомый элемент массива {Rekurs(mass, unknow, 0, mass.Length - 1)}:{mass[Rekurs(mass, unknow, 0, mass.Length - 1)]}");

}

}

catch

{

Console.WriteLine("Введено неправильное число");

}

Console.ReadKey();

}

//Рекурсвиный метод

static int Rekurs(int[]mass,int target,int low,int high)

{

//Если левая часть больше правой возвращаем -1

if(low>high)

{

return -1;

}

//Вычисляем середину

int mid = (low + high) / 2;

//Если элемент середины равен искомому числу

if (target < mass[mid])

{

return Rekurs(mass, target, low, mid - 1);

}

else if(target>mass[mid])

{

return Rekurs(mass, target, mid+1, high);

}

else

{

return mid;

}

}

}

**Вывод:** Разработать рекурсивный алгоритм поиска в отсортированном массиве и оценить его временную сложность в сравнении с итеративным поиском.