Абабий Илья 215гр

**Практическое занятие № 2 Оценка сложности алгоритмов поиска**

**Цель занятия**: **:** научиться оценивать временную сложность алгоритмов поиска в отсортированном массиве.

Вариант 1:

**Ход работы:**

static void Main(string[] args)

{

Write("Введите кол-во элементов в массиве: ");

int a = int.Parse(ReadLine());

int[] nums = new int[a];

Write("Введите искомое число: ");

int n = int.Parse(ReadLine());

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

{

Write($"Введите {i} элемент массива: ");

nums[i] = int.Parse(ReadLine());

}

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

{

int j = i;

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

{

int temp = nums[j];

nums[j] = nums[j-1];

nums[j-1] = temp;

j = j - 1;

}

}

foreach (int i in nums)

{

WriteLine($"mass = {i}");

}

//линейный

int res = 0;

for (int i = 0; i < n; i++)

{

if(nums[i] == n)

{

res = i;

}

}

WriteLine($"номер искомого элемента в массиве - {res}");

//бинарный

int b = Binary(nums, n);//Переменная для бинарного поиска

if (b == -1)

{

WriteLine("Соответствующих элементов в массиве нет");

}

else

{

WriteLine($"nums {b} = {nums[b]}");

}

ReadKey();

}

static int Binary(int[] mass, int m)

{

int left = 0; // левая граница

int right = mass.Length - 1; // правая граница

while (left <= right)

{

int midle = left + (right - left) / 2;

if (mass[midle] == m)

{

return midle;

}

if (mass[midle] < m)

{

left = midle + 1;

}

else

{

right = midle - 1;

}

}

return -1;

}

}**Вывод:** Мы научились оценивать временную сложность алгоритмов поиска в отсортированном массиве.