**Лабораторная работа №3 (Отчет)**

**Марченко Максим 9 группа**

**9 ВАРИАНТ**

Разработать следующий алгоритм и программу с использованием рекурсии. Ввода одномерного массива и линейного поиска целочисленного значения ключа в нем.

**1.Название и цель работы.**

РАЗРАБОТКА РЕКУРСИВНЫХ АЛГОРИТМОВ

***Цель работы:*** Разработка программ, реализующих различные рекурсивные алгоритмы, и оценка их временной и пространственной сложности.

**2. Словесное описание заданного алгоритма поиска.**

Сортируем исходный массив до запуска алгоритма. Находим центральный элемент, если он равен искомому числу, то возвращаем его, если больше, то вызываем рекурсивно поиск по правой части массива и повторяем все тоже самое. Если меньше, то все тоже самое, только с левой части.

**3. Текст программы.**

*start = System.nanoTime();  
int result = RecursionSearchNum(arrayNum,0, arrayNum.length - 1, number);  
end = System.nanoTime();*

*//Функция поиска*

*private static int RecursionSearchNum(int[] arr, int first, int last, int num){  
 if (last >= first) {  
 int mid = first + (last - first) / 2;  
 if (arr[mid] == num) return arr[mid];  
 if (arr[mid] > num) return RecursionSearchNum(arr, first, mid - 1, num);  
 return RecursionSearchNum(arr, mid + 1, last, num);  
 }  
 return -1;  
}*

**4. Формулы верхней оценки временной и емкостной сложности заданного алгоритма.**

***Верхняя оценка алгоритма:*** O().

***Асимптотическая оценка алгоритма для лучшего случая:*** Лучший случай будет тогда, когда нужное слово стоит на центральном элементе, тогда:

Θ(n) = 1(вызов функции RecursionSearchWord) + 1(проверка условия) + 1(инициализация mid) + 3(сумма, разность и деление) + 1(приравнивание mid значения) + 1(проверка условия) + 2(индексация одномерного массива) + 1(возвращение числа) = 11 элементарных операций

***Асимптотическая оценка алгоритма в худшем случае:*** Худший случай будет тогда, когда слово в массиве отсутствует или стоит на последней проверке, тогда:

Θ(n) = 1(вызов функции RecursionSearchWord) + \* (1(проверка условия) + 1(инициализация mid) + 3(сумма, разность и деление) + 1(приравнивание mid значения) + 2(1(проверка условия) + 2(индексация одномерного массива)) + 1(возвращение значения) + 1(вызов функции рекурсивно)) = 1 + 15

**5.Результаты экспериментальной оценки временной и емкостной сложности заданного алгоритма**.

*Время выполнения поиска в лучшем случае: 0.012163 миллисекунд!!*

*Время выполнения поиска в худшем случае: 0.016842 миллисекунд!!*

*для массива размерностью 40*