**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математической кибернетики и информационных технологий

**Отчет по лабораторной работе**

по дисциплине «СИАОД»

на тему: «Алгоритмы сортировки»

Выполнил: студент группы БВТ1802

Сурин В.И.

Руководитель:

Кутейников Иван Алексеевич

Москва 2020

**Цель работы**: Реализовать интерполяционный поиск. Организовать генерацию начального набора случайных данных. Добавить реализацию добавления, поиска и удаления элементов. Оценить время работы каждого алгоритма поиска и сравнить его со временем работы стандартной функции поиска, используемой в выбранном языке программирования.

**Выполнение работы:**

В качестве стандартного метода поиска использовался BinarySearch. Был реализован InterpSearch.

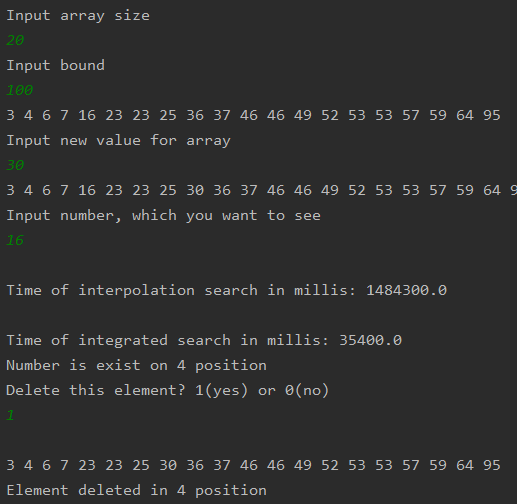
Класс InterpSearch

public class InterpSearch {  
 public static int interpolationSearch(int[] sortedArray, int toFind) {  
 // Возвращает индекс элемента со значением toFind или -1, если такого элемента не существует  
 int mid;  
 int low = 0;  
 int high = sortedArray.length - 1;  
  
 while (sortedArray[low] < toFind && sortedArray[high] > toFind) {  
 mid = low + ((toFind - sortedArray[low]) \* (high - low)) / (sortedArray[high] - sortedArray[low]);  
  
 if (sortedArray[mid] < toFind)  
 low = mid + 1;  
 else if (sortedArray[mid] > toFind)  
 high = mid - 1;  
 else  
 return mid;  
 }  
  
 if (sortedArray[low] == toFind)  
 return low;  
 if (sortedArray[high] == toFind)  
 return high;  
  
 return -1; // Not found  
 }  
}

Класс Main

import java.util.Random;  
import java.util.Scanner;  
import java.util.Arrays;  
  
public class Main {  
 public static void delElement(int[] arr, int index) {  
 for (int i = index; i < arr.length - 1; i++)  
 arr[i] = arr[i + 1];  
 arr[arr.length - 1] = 0;  
 if (arr.length < 30) {  
 for (int i = 0; i < arr.length - 1; i++)  
 System.*out*.print(arr[i] + " ");  
 }  
 }  
  
 public static int[] pushBack(int[] arr, int size, int value) {  
 int[] newArr = new int[size + 1];  
  
 for (int i = 0; i < size; i++)  
 newArr[i] = arr[i];  
 newArr[size] = value;  
  
 return newArr;  
 }  
  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 int n, key, bound;  
 System.*out*.println("Input array size");  
 n = in.nextInt();  
  
 System.*out*.println("Input bound");  
 bound = in.nextInt();  
  
 int[] array = new int[n];  
 Random random = new Random();  
  
 for (int j = 0; j < n; j++) {  
 array[j] = random.nextInt(bound);  
 }  
  
 Arrays.*sort*(array);  
  
 if (array.length < 30) {  
 for (int j = 0; j < n; j++) {  
 System.*out*.print(array[j] + " ");  
 }  
 }  
 else  
 System.*out*.println("The array is very long and will not be output");  
  
 System.*out*.println();  
 System.*out*.println("Input new value for array");  
 int newVal = in.nextInt();  
 int[] newArr = *pushBack*(array, n, newVal);  
 Arrays.*sort*(newArr);  
  
 if (newArr.length < 30) {  
 for (int j = 0; j < newArr.length; j++) {  
 System.*out*.print(newArr[j] + " ");  
 }  
 }  
  
 System.*out*.println();  
 System.*out*.println("Input number, which you want to see");  
 key = in.nextInt();  
  
 double before = System.*nanoTime*();  
 int pos = InterpSearch.*interpolationSearch*(newArr, key);  
 double after = System.*nanoTime*();  
  
 System.*out*.println();  
 System.*out*.println("Time of interpolation search in nanos: " + (after - before));  
  
 before = System.*nanoTime*();  
 int pos2 = Arrays.*binarySearch*(newArr, key);  
 after = System.*nanoTime*();  
  
 System.*out*.println();  
 System.*out*.println("Time of integrated search in nanos: " + (after - before));  
  
 if (pos == -1) {  
 System.*out*.println("Number is not exist");  
 }  
 else {  
 System.*out*.println("Number is exist on " + pos + " position");  
 System.*out*.println("Delete this element? 1(yes) or 0(no)");  
 int dial;  
 dial = in.nextInt();  
 System.*out*.println();  
 if (dial == 1) {  
 *delElement*(newArr, pos);  
 System.*out*.println();  
 System.*out*.println("Element deleted in " + pos + " position");  
 }  
 }  
 }  
}

**Тесты программы:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер  Тип сорт. | 1000 | 10000 | 100000 | 1000000 |
| InterpSearch | 2080400нс | 2110300нс | 2535300нс | 2444200нс |
| BinarySearch | 36200нс | 37500нс | 39600нс | 37500нс |

**Вывод:** в ходе лабораторной работы я изучил алгоритмы поиска в массиве и реализовал интерполяционный поиск. Сравнил время поиска реализованного метода со встроенным бинарным поиском.