|  |  |
| --- | --- |
| Student Name | Uzair Hussain |
| Roll Number | 21SW085 |
| Section # | 3rd or III |
| Lab # | 4th and 5th – Searching in 1D and 2D Arrays |

**Task#01**

**Code:**

import java.util.Scanner;

class Search {

    public static void SingleSearch(int[] arr, int key) {

        for (int i = 0; i < arr.length; i++) {

            if (arr[i] == key) {

                System.out.println("At index : " + i + " , we found number " + key);

                break;

            }

        }

    }

    public static void DoubleSearch(int[][] arr2, int key) {

        for (int i = 0; i < arr2.length; i++) {

            for (int j = 0; j < arr2.length - 1; j++) {

                if (key == arr2[i][j]) {

                    System.out.println("At index: (" + i + " , " + j + "), we found number " + key);

                }

            }

        }

    }

}

class Task1\_LinearSearch {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int[] arr = { 5, 6, 2, 9, 11, 76, 43 };

        int[][] arr2 = { { 23, 65 }, { 54, 32 }, { 2, 8 } };

        Search s = new Search();

        System.out.print("Enter number you want to search in 1D search: ");

        int key1 = sc.nextInt();

        long before1 = System.nanoTime();

        s.SingleSearch(arr, key1);

        long after1 = System.nanoTime();

        System.out.print("Enter number you want to search in 2D search: ");

        int key2 = sc.nextInt();

        long before2 = System.nanoTime();

        s.DoubleSearch(arr2, key2);

        long after2 = System.nanoTime();

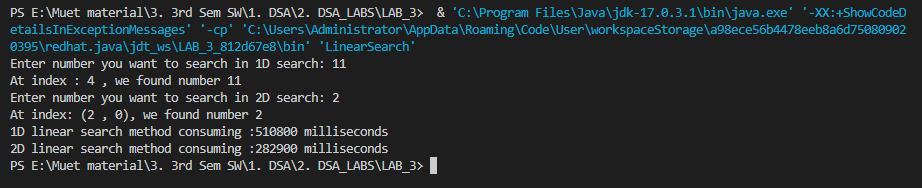
        System.out.println("1D linear search method consuming :" + (after1 - before1) + " milliseconds");

        System.out.println("2D linear search method consuming :" + (after2 - before2) + " milliseconds");

    }

}

**Output:**

****

**Task 02:**

**Code 1:**

import java.util.Arrays;

import java.util.Scanner;

class BinSearch {

    static int SinSearch(int[] a, int x) {

        int p = 0;

        int q = a.length - 1;

        while (p <= q) {

            int i = (p + q) / 2;

            if (x == a[i]) {

                return i;

            } else if (a[i] < x) {

                p = i + 1;

            } else {

                q = i - 1;

            }

        }

        return -5;

    }

}

class Task2\_BinarySearch\_1D {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int[] arr1 = { 5, 6, 2, 9, 11, 76, 43 };

        Arrays.sort(arr1);

        BinSearch b = new BinSearch();

        System.out.print("Enter number you want to search in 1D Binary search: ");

        int key1 = sc.nextInt();

        long before1 = System.nanoTime();

        System.out.println("Found at index: "+b.SinSearch(arr1, key1));

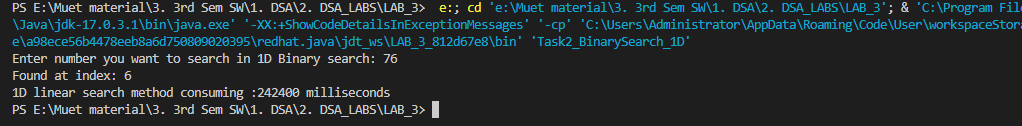
        long after1 = System.nanoTime();

        System.out.println("1D linear search method consuming :" + (after1 - before1)+ " milliseconds");

    }

}

**Output 1:**

****

**Code 2:**

import java.util.Arrays;

import java.util.Scanner;

class Test {

    void binary\_search\_2D(int[][] arr, int target) {

        int row = 0;

        int col = arr[row].length - 1;

        while (row < arr.length && col >= 0) {

            if (arr[row][col] == target) {

                System.out.println("The row index is  " + row + "\n" + "The column index is " + col);

            }

            // Target lies in further row

            if (arr[row][col] < target) {

                row++;

            }

            // Target lies in previous column

            else {

                col--;

            }

        }

    }

}

class Task2\_BinarySearch\_2D {

    public static void main(String[] args) {

        Test t = new Test();

        Scanner sc = new Scanner(System.in);

        int[][] arr2 = { { 1, 2, 3, 4, 5 }, { 6, 7, 8, 9, 10 } };

        System.out.print("Enter number you want to search in 2D search: ");

        int key2 = sc.nextInt();

        long before2 = System.nanoTime();

        t.binary\_search\_2D(arr2, key2);

        long after2 = System.nanoTime();

        System.out.println("2D linear search method consuming :" + (after2 - before2)

                + " milliseconds");

    }

}

**Output 2:**

**Text

Description automatically generated**

**The End!**