**PRACTICAL NO. 1**

Roll no.: 14

Batch: B1

Name: Abhishek Angadi

Date: 6/8/2025 (late admission)

Code:

import java.util.\*;

public class Practical1 {

    int[] randvalues(int max,int min, int size){

            Random r = new Random();

           int[] data = new int[size];

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

            data[i]=r.nextInt((max-min+1))+min;

           }

           return data;

        }

    void TaskA(int[] temp, int[] pressure){

        double start = System.nanoTime();

        int minTemp=Integer.MAX\_VALUE;

        int maxP= Integer.MIN\_VALUE;

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

            if(temp[i]<minTemp){

                minTemp=temp[i];

            }

            if(pressure[i]>maxP){

                maxP=pressure[i];

            }

        }

        double end = System.nanoTime();

        System.out.println("Minimum Temperature through task A: "+minTemp);

        System.out.println("Maximum Temperature through task A: "+maxP);

        System.out.println("Time taken for completion for task A: "+(end-start)/ 1\_000\_000.0);

    }

    void TaskB(int[] temp, int[] pressure){

    double start = System.nanoTime();

    int minTemper = temp[0];

    int maxpressure = pressure[0];

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

        for(int j = 0; j < temp.length; j++){

            if(temp[i] < temp[j]){

                if(temp[i]<minTemper){

                minTemper = temp[i];

                }

            }

        }

    }

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

        for(int j = 0; j < pressure.length; j++){

            if(pressure[i] > pressure[j]){

                if(pressure[i]>maxpressure){

                maxpressure = pressure[i];

                }

            }

        }

    }

    double end = System.nanoTime();

    System.out.println("Minimum Temperature through task B: " + minTemper);

    System.out.println("Maximum Pressure through task B: " + maxpressure);

    System.out.println("Time taken for completion for task B: " + (end - start) / 1\_000\_000.0);

    }

    void TaskC(int[] data){

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

            int temp=0;

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

                if(data[j]>data[j+1]){

                    temp=data[j];

                    data[j]=data[j+1];

                    data[j+1]=temp;

                }

            }

        }

        //Linear search////////////////////////////////

        double start = System.nanoTime();

        int occurence1 = 0;

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

            if(data[i]>=30){

                occurence1=i;

                break;

            }

        }

        double end = System.nanoTime();

        //binary/////////////////////////////////////////////////

        double start1 = System.nanoTime();

        int occurence2=0;

        int low=0;

        int high=data.length-1;

        int mid=(low+high)/2;

        while(low<=high){

            if(data[mid]>=30){

                occurence2=mid;

                high=mid-1;

                mid=(high+low)/2;

            }

            else{

                low=mid+1;

                mid=(high+low)/2;

            }

        }

        double end1 = System.nanoTime();

        System.out.println("Occurence with Linear search: "+occurence1);

        System.out.println("Time for linear search: "+(end-start)/1\_000\_000.0);

        System.out.println("Occurence wiith binary search: "+ occurence2);

        System.out.println("Time for binary search: "+(end1-start1)/1\_000\_000.0);

    }

    public static void main(String[] args) {

        Practical1 p = new Practical1();

        int[] temp=p.randvalues(50, -20, 100);

        int[] pressure = p.randvalues(1050, 950,100);

        p.TaskA(temp, pressure);

        System.out.println();

        System.out.println();

        p.TaskB(temp, pressure);

        System.out.println();

        System.out.println();

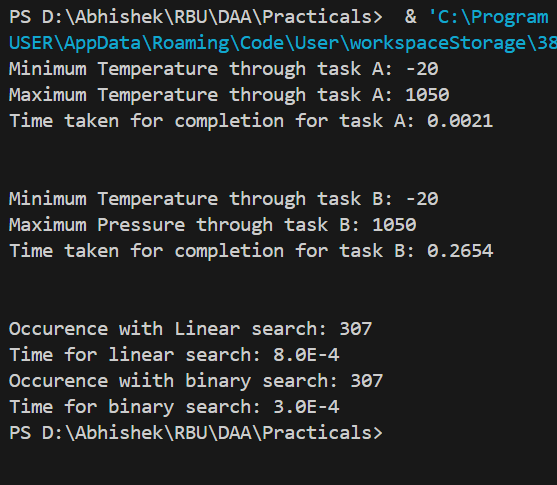
        int[] newtemp = p.randvalues(50, 20,1000);

        p.TaskC(newtemp);

    }

}

Output:



Conclusion:

Hence, we successfully performed this practical and analyzed Time and complexity of loops for a sensor data monitoring system by generating random sensor readings such as temperature, and pressure. We successfully analyzed and compared the performance of different algorithms

Github: https://github.com/Shadow3456rh/DAA-Rbu-Practicals