12-11-24

DSA PRACTISE 3

anagram program

public class AnagramProgram {

    public static Boolean isAnagram(String s , String t){

        int [] count = new int[26];

        for(char c:s.toCharArray()){

            count[c - 'a']++;

        }

        for(char c:t.toCharArray()){

            count[c - 'a']--;

        }

        for(int i=0;i<26;i++){

            if(count[i]!=0){

                return false;

            }

        }return true;

    }

    public static void main(String[] args){

        String s = "code";

        String t = "deco";

        System.out.println("Given String are Anagram? "+ isAnagram(s,t));

    }

}

row with max 1s'

public class RowWithMax1s {

    public static int Max1(int [][] matrix){

        int n = matrix.length;

        int m = matrix[0].length;

        int maxrowindex = -1;

        int col = n-1;

        int row = 0;

        while(row<n&&col>=0){

            if (matrix[row][col]==1){

                maxrowindex = row ;

                col --;

            }

            else{

                row++;

            }

        }return maxrowindex;

    }

    public static void main(String[] args) {

        int[][] matrix = {

            {0, 0, 0, 1},

            {0, 1, 1, 1},

            {1, 1, 1, 1},

            {0, 0, 0, 0}

        };

        int result = Max1(matrix);

        if (result != -1) {

            System.out.println("Row with maximum 1s: " + result);

        } else {

            System.out.println("No 1s found in the matrix");

        }

    }

    }

Longest consequtive subsequence

import java.util.HashSet; // Line 1

import java.util.Set;

public class LongestConsecutiveSeq {

    public static int ConsecutiveSeq(int[] nums){

        Set<Integer> set = new HashSet<>();

        for(int n :nums){

            set.add(n);

        }

        int max=0;

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

            int num = nums[i];

            if(set.contains(num-1))continue;

            int currlength=0;

            while (set.contains(num)){

                currlength+=1;

                num += 1;

            }

            max = Math.max(currlength,max);

        }return max;

    }

    public static void main(String[] args){

        int[] nums = {100,4,200,1,3,2};

        int result = ConsecutiveSeq(nums);

        System.out.println("Length of consecutive numbers is "+ result);

    }

}