11-11-2024

Triplet sum in array

import java.util.Arrays;

public class TripleSum {

    public static int[] Triples(int arr[],int sum){

        Arrays.sort(arr);

        for (int i = 0; i < arr.length -2;i++ ){

            int k = arr.length-1;

            int j = i+1;

            int s = sum - arr[i];

            while(j<k){

                if (arr[j]+arr[k]==s){

                    return new int[]{arr[i],arr[j],arr[k]};

                }

                if (arr[j]+arr[k]<s){

                    j++;

                }else{

                    k--;

                }

            }

        }

        return null;

    }

    public static void main(String[] args){

        int[] arr = {1,0,-1,3,2};

        int sum = 3;

        int[] result = Triples(arr, sum);

        if (result != null){

        System.out.println("Triple Sum: " + Arrays.toString(result));

        }

        else{

            System.out.println("No Triplets found");

        }

    }

}

Floor in sorted array

public class Floorelement {

    public static int sortedarray(int[]arr,int key){

        int start = 0;

        int end = arr.length-1;

        int res = -1;

        while(start<=end){

            int mid = start+((end-start)/2);

            if (arr[mid]==key){

                return arr[mid];

            }

            if (arr[mid]< key){

                res = arr[mid];

                start = mid +1;

            }

            else{

                if(arr[mid]>key){

                    res = arr[mid];

                    end = mid - 1;

                }

            }

        }return res;

    }

    public static void main(String[] args){

        int[] arr = {1,2,3,4,8,10,10,12,19};

        int key = 5;

        int result = sortedarray(arr, key);

        System.out.println("Floor of the key is: " + result);

    }

}

Check equal arrays

public class Check2Arrays {

    public static boolean check(longA[],long B[],int N){

        Map<long,Integer>counterMap = new HashMap<>();

        for(int index=0;index<N;index++){

            int counter=counterMap.getOrDefault(A[index],0);

            counterMap.put(A[index],counter+1);

        }

        for(int index = o ; index<N;index++){

            int counter = counterMap.getOrDefault(B[index],0);

            if (counter==0){

                return false;

            }

            counterMap.put(B[index],counter-1);

        }

    }

    return true;

}

Palindrome linked list