DSA PRACTICE PROBLEM DAY 4

PROBLEM 1:

Given an array **arr[]** and an integer **k** where k is smaller than the size of the array, the task is to find the **kth smallest** element in the given array.

**Follow up:** Don't solve it using the inbuilt sort function.

CODE :

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the Size of Array :");

int n = scan.nextInt();

System.out.println("Enter the Elements :");

int[] arr = new int[n];

for(int i=0;i<n;i++){

arr[i] = scan.nextInt();

}

System.out.println("Enter the k value :");

int k = scan.nextInt();

PriorityQueue<Integer> pq = new PriorityQueue<>((a, b) -> b - a);

for (int i : arr) {

pq.offer(i);

if (pq.size() > k) {

pq.poll();

}

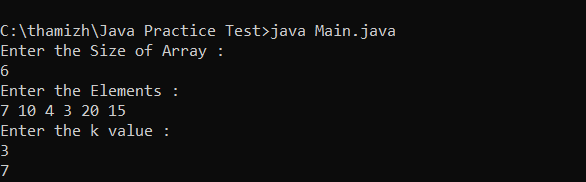
}

System.out.println(pq.peek());

}

}

OUTPUT :



PROBLEM 2 :

MINIMIZE THE HEIGHT 2

CODE:

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the Size of Array :");

int n = scan.nextInt();

System.out.println("Enter the Elements :");

int[] arr = new int[n];

for(int i=0;i<n;i++){

arr[i] = scan.nextInt();

}

System.out.println("Enter the k value :");

int k = scan.nextInt();

Arrays.sort(arr);

int ans = arr[n-1] - arr[0];

int tempmin = arr[0];

int tempmax = arr[n-1];

for(int i=1;i<n;i++) {

if(arr[i] - k < 0) {

continue;

}

tempmin = Math.min(arr[0] + k, arr[i] - k);

tempmax = Math.max(arr[i-1] + k , arr[n-1] - k);

ans = Math.min(ans, tempmax - tempmin);

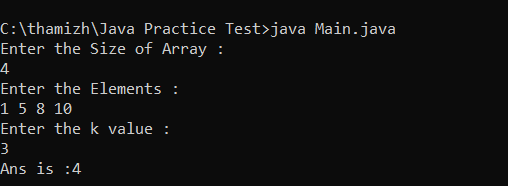
}

System.out.println("Ans is :" + ans);

}

}

OUTPUT :



PROBLEM 3 :

Parenthesis Checker

CODE :

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the String :");

String s = scan.next();

Stack<Character> st = new Stack<>();

int n = s.length();

boolean flag = true;

for(char ch : s.toCharArray()){

if(ch=='(' || ch=='{' || ch=='['){

st.push(ch);

}else{

if(st.isEmpty() || (ch==')' && st.peek()!='(') || (ch=='}' && st.peek()!='{') || (ch==']' && st.peek()!='[')){

flag = false;

break;

}else{

st.pop();

}

}

}

if(flag){

System.out.println(st.isEmpty()?"VALID":"NOT VALID");

}else{

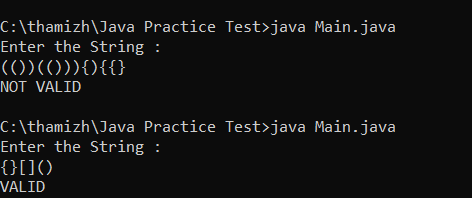
System.out.println("NOT VALID");

}

}

}

OUTPUT :



PROBLEM 4:

EQUILIBRIUM POINT

CODE:

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the Size of the Array :");

int n = scan.nextInt();

int[] arr = new int[n];

System.out.println("Enter the elements in the array :");

for(int i=0;i<n;i++){

arr[i] = scan.nextInt();

}

int[] prefix = new int[n];

int[] suffix = new int[n];

prefix[0] = arr[0];

suffix[n-1] = arr[n-1];

for(int i=1;i<n;i++){

prefix[i] = prefix[i-1] + arr[i];

suffix[n-i-1] = suffix[n-i] + arr[n-i-1];

}

int position = -1;

for(int i=0;i<n;i++){

if(prefix[i] == suffix[i]){

position = i+1;

break;

}

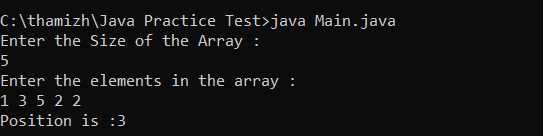
}

System.out.println("Position is :" + position);

}

}

OUTPUT:



PROBLEM 5:

BINARY SEARCH:

CODE:

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the Size of the Array :");

int n = scan.nextInt();

int[] arr = new int[n];

System.out.println("Enter the elements in the array :");

for(int i=0;i<n;i++){

arr[i] = scan.nextInt();

}

System.out.println("Enter the Number for Search :");

int x = scan.nextInt();

Arrays.sort(arr);

int low = 0, high = arr.length - 1;

int result = -1;

while (low <= high) {

int mid = low + (high - low) / 2;

if (arr[mid] == x){

result = mid;

break;

}

if (arr[mid] < x)

low = mid + 1;

else

high = mid - 1;

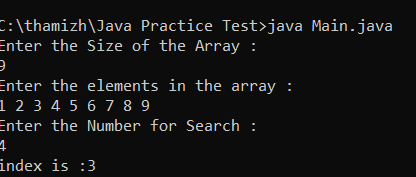
}

System.out.println("index is :" + result);

}

}

OUTPUT :



PROBLEM 6 :

UNION OF TWO ARRAY WITH DUPLICATE ELEMENTS

CODE:

import java.util.\*;

public class Main {

public static void main(String... argv) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter the Size of the Array1 :");

int n = scan.nextInt();

System.out.println("Enter the Size of the Array2 :");

int m = scan.nextInt();

int[] a = new int[n];

int[] b = new int[m];

System.out.println("Enter the elements in the array1 :");

for(int i=0;i<n;i++){

a[i] = scan.nextInt();

}

System.out.println("Enter the elements in the array2 :");

for(int i=0;i<m;i++){

b[i] = scan.nextInt();

}

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

for(int i:a){

set.add(i);

}

for(int i : b){

set.add(i);

}

System.out.println("result is :"+ set.size());

}

}

OUTPUT:

