### Print Binary Tree levels in sorted order

Given an array **arr**[] which contains data of **N** nodes of Complete Binary tree in level order fashion. The task is to print the level order traversal in sorted order.

**Example 1:**

**Input:**

N = 7

arr[] = {7 6 5 4 3 2 1}

**Output:**

7

5 6

1 2 3 4

**Explanation:** The formed Binary Tree is:

7

/ \

6 5

/ \ / \

4 3 2 1

**Example 2:**

**Input:**

N = 6

arr[] = {5 6 4 9 2 1}

**Output:**

5

4 6

1 2 9

**Explanation:** The formed Binary Tree is:

5

/ \

6 4

/ \ /

9 2 1

//{ Driver Code Starts

//Initial Template for Java

import java.io.\*;

import java.util.\*;

class CodingMaxima

{

public static void main (String[] args)

{

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while(t-- > 0)

{

int n = sc.nextInt();

int[] arr = new int[n];

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

arr[i] = sc.nextInt ();

ArrayList <ArrayList <Integer>> res = new Solution().binTreeSortedLevels (arr, n);

for (int i = 0; i < res.size(); i++)

{

for (int j = 0; j < (res.get (i).size()); j++)

{

System.out.print (res.get(i).get(j) + " ");

}

System.out.println ();

}

}

}

}

// Contributed By: Pranay Bansal

// } Driver Code Ends

//User function Template for Java

class Solution

{

public static ArrayList <ArrayList <Integer>> binTreeSortedLevels (int arr[], int n)

{

ArrayList <ArrayList <Integer>> ans=new ArrayList <ArrayList <Integer>>();

if(n==0)

return ans;

Queue<Integer> q=new LinkedList<>();

q.add(arr[0]);

int j=0;

while(!q.isEmpty()){

int size=q.size();

ArrayList <Integer> ar=new ArrayList <Integer>();

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

int temp=q.poll();

ar.add(temp);

if(j\*2+1<n)

q.add(arr[j\*2+1]);

if(j\*2+2<n)

q.add(arr[j\*2+2]);

j++;

}

Collections.sort(ar);

ans.add(ar);

}

return ans;

}

}