**Wave Array: -**

**Easy** Accuracy: **63.69%** Submissions: **197K+** Points: **2**

Given a **sorted** array **arr[]** of distinct integers. Sort the array into a wave-like array(In Place).  
In other words, arrange the elements into a sequence such that arr[1] >= arr[2] <= arr[3] >= arr[4] <= arr[5].....

If there are multiple solutions, find the lexicographically smallest one.

**Note:**The given array is sorted in ascending order, and you don't need to return anything to make changes in the original array itself.

**Example 1:**

**Input:**

n = 5

arr[] = {1,2,3,4,5}

**Output:** 2 1 4 3 5

**Explanation:** Array elements after

sorting it in wave form are

2 1 4 3 5.

**Example 2:**

**Input:**

n = 6

arr[] = {2,4,7,8,9,10}

**Output:** 4 2 8 7 10 9

**Explanation:** Array elements after

sorting it in wave form are

4 2 8 7 10 9.

**Your Task:**  
The task is to complete the function **convertToWave**(), which converts the given array to a wave array.

**Expected Time Complexity:**O(n).  
**Expected Auxiliary Space:**O(1).

**Constraints:**  
1 ≤ n ≤ 106  
0 ≤ arr[i] ≤107

**Code: -**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// #include <algorithm>

// } Driver Code Ends

class Solution{

public:

// arr: input array

// n: size of array

//Function to sort the array into a wave-like array.

void convertToWave(int n, vector<int>& arr){

for(int i=0; i<n; i+=2){

if(i+1 < n)

swap(arr[i], arr[i+1]);

}

return;

}

};

//{ Driver Code Starts.

int main()

{

int t,n;

cin>>t; //Input testcases

while(t--) //While testcases exist

{

cin>>n; //input size of array

vector<int> a(n); //declare vector of size n

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

cin>>a[i]; //input elements of array

sort(a.begin(),a.end());

Solution ob;

ob.convertToWave(n, a);

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

cout<<a[i]<<" "; //print array

cout<<endl;

}

}

// } Driver Code Ends

**T.C: - O(N)**

**S.C: - O(1)**