### Quick Sort

Quick Sort is a Divide and Conquer algorithm. It picks an element as a pivot and partitions the given array around the picked pivot.  
Given an array arr[], its starting position is low (the index of the array) and its ending position is high(the index of the array).

**Note**: The **low** and **high** are inclusive.

Implement the partition() and quickSort() functions to sort the array.

**Example 1:**

**Input:**

N = 5

arr[] = { 4, 1, 3, 9, 7}

**Output:**

1 3 4 7 9

**Example 2:**

**Input:**

N = 9

arr[] = { 2, 1, 6, 10, 4, 1, 3, 9, 7}

**Output:**

1 1 2 3 4 6 7 9 10

**Expected Time Complexity:** O(N\*logN)  
**Expected Auxiliary Space:** O(logN)

**Constraints:**  
1 <= N <= 103  
1 <= arr[i] <= 104

**Company Tags**

[**VMWare**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=VMWare) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Samsung**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Samsung) [**Hike**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Hike)[**Ola Cabs**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Ola%20Cabs) [**Goldman Sachs**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Goldman%20Sachs) [**Adobe**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Adobe) [**SAP Labs**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=SAP%20Labs) [**Qualcomm**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Qualcomm) [**HSBC**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=HSBC) [**Grofers**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Grofers) [**Target Corporation**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Target%20Corporation)

**Topic Tags**

[**Divide and Conquer**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Divide%20and%20Conquer) [**Sorting**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Sorting) [**Algorithms**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Algorithms)

### Java Code

//{ Driver Code Starts

import java.util.\*;

class Sorting

{

static void printArray(int arr[])

{

int n = arr.length;

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

System.out.print(arr[i]+" ");

System.out.println();

}

// Driver program

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();

new Solution().quickSort(arr,0,n-1);

printArray(arr);

T--;

}

} }

// } Driver Code Ends

class Solution

{

//Function to sort an array using quick sort algorithm.

static void quickSort(int arr[], int low, int high)

{

if(low < high){

int p = partition(arr, low, high);

quickSort(arr, low, p-1);

quickSort(arr, p+1, high);

}

}

static int partition(int arr[], int low, int high)

{

int pivot = arr[high];

int i = low-1, j, temp = 0;

for(j=low;j<high;j++){

if(arr[j]<pivot){

i++;

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

temp = arr[i+1];

arr[i+1] = arr[high];

arr[high] = temp;

return i+1;

}

}