‹ Sorting Challenges

Quicksort 1 - Partition 🔖

H by **HackerRank**

| Problem | Submissions | Leaderboard | Discussions |

The previous challenges covered Insertion Sort, which is a simple and intuitive sorting algorithm with an average case performance of $O(n^2)$. In these next few challenges, we're covering a *divide-and-conquer* algorithm called Quicksort (also known as *Partition Sort*).

Step 1: Divide
Choose some pivot element, p, and partition your unsorted array, ar, into three smaller arrays: $left$, $right$, and $equal$, where each element in $left < p$, each element in $right > p$, and each element in $equal = p$.

Challenge
Given ar and $p = ar[0]$, partition ar into $left$, $right$, and $equal$ using the *Divide* instructions above. Then print each element in $left$ followed by each element in $equal$, followed by each element in $right$ on a single line. Your output should be space-separated.

Note: There is no need to sort the elements in-place; you can create two lists and stitch them together at the end.

Input Format

The first line contains n (the size of ar).
The second line contains n space-separated integers describing ar (the unsorted array). The first integer (corresponding to $ar[0]$) is your pivot element, p.

Constraints

- $1 \le n \le 1000$

- $-1000 \le x \le 1000, x \in ar$

- All elements will be unique.

- Multiple answer can exists for the given test case. Print any one of them.

Output Format

On a single line, print the partitioned numbers (i.e.: the elements in $left$, then the elements in $equal$, and then the elements in $right$). Each integer should be separated by a single space.

Sample Input

```
5
4 5 3 7 2
```

Sample Output

```
3 2 4 5 7
```

Explanation

$ar = [4, 5, 3, 7, 2]$
Pivot: $p = ar[0] = 4$.
$left = \{\}$; $equal = \{4\}$; $right = \{\}$

$ar[1] = 5 \geq p$, so it's added to *right*.
$left = \{\}; equal = \{4\}; right = \{5\}$

$ar[2] = 3 < p$, so it's added to *left*.
$left = \{3\}; equal = \{4\}; right = \{5\}$

$ar[3] = 7 \geq p$, so it's added to *right*.
$left = \{3\}; equal = \{4\}; right = \{5, 7\}$

$ar[4] = 2 < p$, so it's added to *left*.
$left = \{3, 2\}; equal = \{4\}; right = \{5, 7\}$

We then print the elements of *left*, followed by *equal*, followed by *right*, we get: `3 2 4 5 7`.

This example is only one correct answer based on the implementation shown, but it is not the only correct answer (e.g.: another valid solution would be `2 3 4 5 7`).

f   🐦   in

Solved score: 10.00pts

Submissions: 38340
Max Score: 10
Difficulty: Easy

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable) 🔀 🕓   C#   ⌄   ⤢ ⚙

```csharp
 5  using System;
 6  using System.Collections.Generic;
 7  using System.IO;
 8  class Solution {
 9  static void partition(int[] ar) {
10
11      int pivote = ar[0];
12          List<int> left = new List<int>();
13          List<int> equal = new List<int>();
14          List<int> rigth = new List<int>();
15
16          equal.Add(pivote);
17
18          for (int i = 1; i < ar.Length; i++)
19          {
20              if (ar[i] < pivote)
21              {
22                  left.Add(ar[i]);
23              }
24              else if (ar[i] == pivote)
25              {
26                  equal.Add(ar[i]);
27              }
28              else if (ar[i] > pivote)
29              {
30                  rigth.Add(ar[i]);
31              }
32          }
33
34          int index = 0;
35          for (int i = 0; i < left.Count; i++)
36          {
37              ar[index++] = left[i];
38          }
39          for (int i = 0; i < equal.Count; i++)
40          {
41              ar[index++] = equal[i];
42          }
```

```
43              for (int i = 0; i < rigth.Count; i++)
44              {
45                  ar[index++] = rigth[i];
46              }
47
48
49
50     }
51     /* Tail starts here */
52         static void Main(String[] args) {
53
54             int _ar_size;
55             _ar_size = Convert.ToInt32(Console.ReadLine());
56             int [] _ar =new int [_ar_size];
57             String elements = Console.ReadLine();
58             String[] split_elements = elements.Split(' ');
59             for(int _ar_i=0; _ar_i < _ar_size; _ar_i++) {
60                 _ar[_ar_i] = Convert.ToInt32(split_elements[_ar_i]);
61             }
62
63             partition(_ar);
64
65             for (int i = 0; i < _ar.Length; i++)
66              {
67                 Console.Write(_ar[i] + " ");
68              }
69
70         }
71     }
72
```

Line: 67 Col: 36

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

---

Congrats, you solved this challenge!

✔ Test Case #0   ✔ Test Case #1   ✔ Test Case #2
✔ Test Case #3   ✔ Test Case #4

Next Challenge

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature