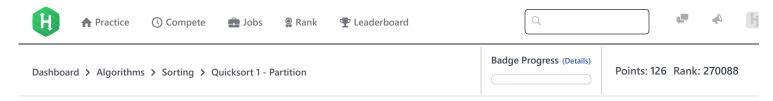
15/11/2017 HackerRank



# Quicksort 1 - Partition



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

#### **Sample Output**

3 2 4 5 7

## **Explanation**

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

15/11/2017 HackerRank

```
ar[1] = 5 \ge 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 \ge 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:47393
Max Score:10
Difficulty: Easy
Rate This Challenge:
☆☆☆☆☆

```
Current Buffer (saved locally, editable) & 49
                                                                                               Java 7
                                                                                                                                  Ö
 1 ▼ import java.util.*;
 2 ▼ public class Solution {
 3
 4 ▼
                static void partition(int[] ar) {
 5
            }
 6
 7
 8 ▼
      static void printArray(int[] ar) {
 9 🔻
              for(int n: ar){
10
                  System.out.print(n+" ");
11
                 System.out.println("");
12
           }
13
14
15 🔻
           public static void main(String[] args) {
16
                 Scanner in = new Scanner(System.in);
                 int n = in.nextInt();
17
18 ▼
                 int[] ar = new int[n];
19 ▼
                 for(int i=0;i<n;i++){</pre>
20 1
                    ar[i]=in.nextInt();
21
22
                 partition(ar);
23
            }
24
        }
25
                                                                                                                         Line: 1 Col: 1
                                                                                                            Run Code
                                                                                                                          Submit Code
                       Test against custom input
Upload Code as File
```

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

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