

# Day 20: Sorting



Problem Submissions Leaderboard Discussions Editorial Tutorial

#### Objective

Today, we're discussing a simple sorting algorithm called Bubble Sort. Check out the Tutorial tab for learning materials and an instructional video!

Consider the following version of Bubble Sort:

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

  for (int j = 0; j < n - 1; j++) {
     if (a[j] > a[j + 1]) {
        swap(a[j], a[j + 1]);
        numberOfSwaps++;
     }
  }

  if (numberOfSwaps == 0) {
     break;
  }
}
```

#### Task

Given an array, a, of size n containing distinct elements  $a[0], a[1], \ldots, a[n-1]$ , sort array a in ascending order using the *Bubble Sort* algorithm above. Once sorted, print the following a lines:

- 1. Array is sorted in numSwaps swaps.
- where *numSwaps* is the number of swaps that took place.
- 2. **First Element:** *firstElement* where *firstElement* is the *first* element in the sorted array.
- 3. Last Element: lastElement where lastElement is the last element in the sorted array.

Hint: To complete this challenge, you will need to add a variable that keeps a running tally of all swaps that occur during execution.

# **Input Format**

The first line contains an integer, n, denoting the number of elements in array a. The second line contains n space-separated integers describing a, where the  $i^{th}$  integer is a[i],  $\forall i \in [0, n-1]$ .

#### Constraints

- $2 \le n \le 600$
- $1 \le a[i] \le 2 \times 10^6$ ,  $\forall i \in [0, n-1]$

#### **Output Format**

There should be 3 lines of output:

1. Array is sorted in numSwaps swaps.

where *numSwaps* is the number of swaps that took place.

2. First Element: firstElement

where *firstElement* is the *first* element in the sorted array.

3. Last Element: lastElement

where *lastElement* is the *last* element in the sorted array.

## Sample Input 0

3 1 2 3

#### Sample Output 0

```
Array is sorted in 0 swaps.
First Element: 1
Last Element: 3
```

## Sample Input 1

3 3 2 1

## Sample Output 1

```
Array is sorted in 3 swaps.
First Element: 1
Last Element: 3
```

# Explanation

## Sample Case 1:

The array is already sorted, so  $\mathbf{0}$  swaps take place and we print the necessary  $\mathbf{3}$  lines of output shown above.

#### Sample Case 2:

The array is not sorted, and its initial values are: {3, 2, 1}. The following 3 swaps take place:

- 1.  $\{3,2,1\} \rightarrow \{2,3,1\}$
- 2.  $\{2, 3, 1\} \rightarrow \{2, 1, 3\}$
- 3.  $\{2, 1, 3\} \rightarrow \{1, 2, 3\}$

At this point the array is sorted and we print the necessary 3 lines of output shown above.

Submissions: 5268

Max Score: 30

Difficulty: Easy

More

```
Current Buffer (saved locally, editable)  
Python 2

#!/bin/python

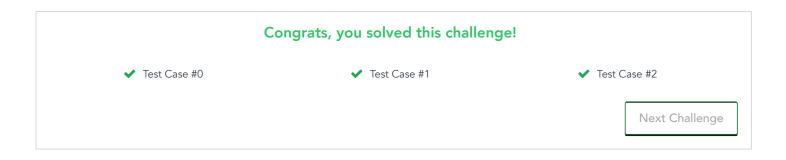
import sys

n = int(raw_input().strip())
```

```
a = map(int,raw_input().strip().split(' '))
 8 numSwaps=0
 9 ▼for i in range(n):
10 🔻
        for j in range(n-1):
            if a[j]>a[j+1]:
11
                temp=a[j+1]
12
13
                a[j+1]=a[j]
                a[j]=temp
14
15
                numSwaps=numSwaps+1
16
        #if numSwaps==0:
17
        #
             break
    print 'Array is sorted in %i swaps.' %numSwaps
18
    print 'First Element:',a[0]
19
   print 'Last Element:', a[len(a)-1]
                                                                                                          Line: 9 Col: 19

    Test against custom input

                                                                                                 Run Code
                                                                                                             Submit Code
1 Upload Code as File
```



Copyright © 2016 HackerRank. All Rights Reserved

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