

Insertion Sort - Part 1

Problem Statement

Sorting

One common task for computers is to sort data. For example, people might want to see all their files on a computer sorted by size. Since sorting is a simple problem with many different possible solutions, it is often used to introduce the study of algorithms.

Insertion Sort

These challenges will cover Insertion Sort, a simple and intuitive sorting algorithm. We will first start with an already sorted list.

Insert element into sorted list

Given a sorted list with an unsorted number V in the right-most cell, can you write some simple code to insert V into the array so it remains sorted?

Print the array every time a value is shifted in the array until the array is fully sorted. The goal of this challenge is to follow the correct order of insertion sort.

Guideline: You can copy the value of V to a variable, and consider its cell "empty". Since this leaves an extra cell empty on the right, you can shift everything over until V can be inserted. This will create a duplicate of each value, but when you reach the right spot, you can replace a value with V .

Input Format

There will be two lines of input:

- s - the size of the array
- ar - the sorted array of integers

Output Format

On each line, output the entire array every time an item is shifted in it.

Constraints

$$1 \leq s \leq 1000$$

$$-10000 \leq x \leq 10000, x \in ar$$

Sample Input

```
5
2 4 6 8 3
```

Sample Output

```
2 4 6 8 8
2 4 6 6 8
2 4 4 6 8
2 3 4 6 8
```

Explanation

3 is removed from the end of the array.

In the 1st line $8 > 3$, 8 is shifted one cell right.
In the 2nd line $6 > 3$, 6 is shifted one cell right.
In the 3rd line $4 > 3$, 4 is shifted one cell right.
In the 4th line $2 < 3$, 3 is placed at position 2.

Task

Complete the method `insertionSort` which takes in 1 parameter:

- *ar* - an array with the value *V* in the right-most cell.

Next Challenge

In the [next Challenge](#), we will complete the insertion sort itself!