

## Remove the minimum

Instructions

Output

- ### The museum of incredible dull things

The museum of incredible dull things wants to get rid of some exhibitions. Miriam, the interior architect, comes up with a plan to remove the most boring exhibitions.

She gives them a rating, and then removes the one with the lowest rating.

However, just as she finished rating all exhibitions, she's off to an important fair, so she asks you to write a program that tells her the ratings of the items after one removed the lowest one. Fair enough.

### Task

Given an array of integers, remove the smallest value. **Do not mutate the original array/list**. If there are multiple elements with the same value, remove the one with a lower index. If you get an empty array/list, return an empty array/list.

Don't change the order of the elements that are left.

### Examples

```
removeSmallest([1,2,3,4,5]) = [2,3,4,5]
removeSmallest([5,3,2,1,4]) = [5,3,2,4]
• removeSmallest([2,2,1,2,1]) = [2,2,2,1]
```

FUNDAMENTALS LISTS DATA STRUCTURES ARRAYS

Solution:

```
function removeSmallest(numbers) {
  throw "TODO: removeSmallest";
}
```

1  
2  
3

```
}
```

## Sample Tests:

```
Test.describe("removeSmallest", function() {  
  
  Test.it("works for the examples", function() {  
  
    Test.assertSimilar(removeSmallest([1, 2, 3, 4, 5]), [2, 3, 4, 5],  
"Wrong result for [1, 2, 3, 4, 5]");  
  
    Test.assertSimilar(removeSmallest([5, 3, 2, 1, 4]), [5, 3, 2, 4],  
"Wrong result for [5, 3, 2, 1, 4]");  
  
    Test.assertSimilar(removeSmallest([2, 2, 1, 2, 1]), [2, 2, 2, 1],  
"Wrong result for [2, 2, 1, 2, 1]");  
  
    Test.assertSimilar(removeSmallest([]), [], "Wrong result for []");  
  
  });  
  
  Test.it("returns [] if the list has only one element", function() {  
  
    for (let i = 0; i < 10; ++i) {  
  
      let x = ~~(Math.random() * 400);  
  
      Test.assertSimilar(removeSmallest([x]), [], `Wrong result for  
${[x]}`);  
  
    }  
  
  });  
  
  function randomArray(length) {  
  
    return Array.from({length: length}, () => ~~(Math.random() * 400));  
  
  }  
  
}
```

```
}
19
20
Test.it("returns a list that misses only one element", function() {
21
    for(let i = 0; i < 10; ++i) {
22
        let arr = randomArray(~~(Math.random() * 10) + 1);
23
        let l = arr.length;
24
        Test.assertSimilar(removeSmallest(arr).length, l - 1, `Wrong result
for ${arr}`);
25
    }
26
});
27
});
```