

QUIZ

Finish this `swap()` function:

```
const swap = (arr, indexOne, indexTwo) => {  
  const  =  ;  
  arr[indexTwo] = arr[indexOne];  
   =  ;  
};
```



You got it!

What array would bubble sort be able to sort in a worst-case time of $O(n^2)$?

[4, 3, 2, 1]



Yes, that's right! Because this array is sorted from largest to smallest, bubble sort would have a runtime of $O(n^2)$.

[1, 2, 3, 4]

What's wrong with this implementation of bubble sort?

```
const bubbleSort = input => {  
  let swapping = true;  
  
  while (swapping) {  
    swapping = false;  
    for (let i = 0; i < input.length - 1; i++) {  
      if (input[i] < input[i + 1]) {  
        swap(input, i, i + 1);  
        swapping = true;  
      }  
    }  
  }  
}  
  
return input;  
};
```

It would only bubble up one number in the array to its correct position

It would sort the elements in the input array from largest to smallest



That's right! The comparator should be `>` in the `if` statement.

How many swaps would it take for bubble sort to sort this array?

[4, 3, 2, 1]

6 swaps



Excellent! Yes, bubble sort would need to swap elements in this array 6 times to get it in order from smallest to largest.

4 swaps

3 swaps

2 swaps

What array would bubble sort be able to sort with the least swaps or in $O(n)$ time?

[4, 3, 2, 1]

[2, 1, 3, 4]

[3, 2, 1]

[1, 2, 3, 4]



Correct! No swaps are needed since this array is already sorted from smallest to largest. So, we can say that if given this input, bubble sort would run in $O(n)$ time.

Finish this bubble sort implementation below so that it continually checks for unsorted elements and swaps them, until the array is completely sorted:

```
const bubbleSort = input => {  
  let swapping = true ;  
  
  while ( swapping ) {  
    swapping = false ;  
  
    for (let i = 0; i < input.length - 1; i++) {  
      if (input[i] > input[i + 1]) {  
        swap(input, i, i + 1);  
        swapping = true ;  
      }  
    }  
  }  
  return input;  
};
```



You got it!

Put the steps of the bubble sort algorithm in order:

```
while array is not sorted  
  for each value in array  
    if currentVal > nextVal  
      swap currentVal and nextVal  
  return array
```



You got it!

If we ran this code, how many times would this implementation of bubble sort loop over the entire input array?

```
const bubbleSort = input => {  
  let swapping = true;  
  
  while (swapping) {  
    swapping = false;  
  
    for (let i = 0; i < input.length - 1; i++) {  
      if (input[i] > input[i + 1]) {  
        swap(input, i, i + 1);  
      }  
    }  
  }  
  
  return input;  
};  
  
bubbleSort([3, 1, 2, 5]);
```

4 times

1 time



Correct! This code will only loop over the input array once because we set our `while` condition to `false` and didn't reset it back to `true` after swapping.