Compare

5 min

We have a loop to run our

Preview: Docs Loading link description

algorithm

, and another that visits each element in the input

Preview: Docs Loading link description

array

but if we were to run this as is, it would only iterate through the array once.

Let's add some additional logic to our code in the **bubbleSort.js** file that will compare neighboring elements and swap them if necessary. For this exercise, you'll only be adding code inside of the for loop of bubbleSort().

Instructions

1. Checkpoint 1 Passed

1.

Inside of the for loop we added, create a conditional that checks if the element at the current index is greater than the element one index after it.

2. Checkpoint 2 Passed

2.

Let's begin adding the logic for the swapping action. To see how we're changing our input array by swapping elements, add the following inside of the if statement you created:

console.log(`Swapping pair ${input[i]}, ${input[i+1]} in [${input}]`);

This code will log a message for every swap made when we execute bubbleSort().

Hint

String interpolation allows us to replace placeholders in a string with values, like the input array or elements stored in the input array.

3. Checkpoint 3 Passed

3.

Swap unordered pairs. After our logging statement, add a call to swap(), the helper function that handles changing the position of pairs of elements. Take a look at the parameters of swap() in the **swap.js** file to see what arguments you need to call it with.

Hint

Make sure you are passing in the arguments to swap() in the correct order.

4. Checkpoint 4 Passed

4.

If we make a swap, we want to loop through the array again to see if we need to make additional swaps to continue “bubbling up” elements in the wrong position.

Keep our while loop running by changing the value of the while condition variable so that it evaluates as true.

Hint

We’re using swapping to store our while condition. To restart the loop, we need to change the value stored in swapping so that the condition evaluates as true.

bubbleSort.js

```
const swap = require('./swap');


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]) {
    console.log(`Swapping pair ${input[i]}, ${input[i+1]} in [${input}]`);
    swap(input, i, i + 1)
    swapping = true;
   }
  }
 }
 return input;
};
```

```
module.exports = bubbleSort;
```

swap.js

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

module.exports = swap;
```