



Algorithm — quickSort



Justinpaoletta [Follow](#)
Jun 30 · 3 min read



Solved with javascript

ALGORITHM SERIES



CHALLENGE (quickSort)

You must write a function that :

- Takes in an array as its input.
- Picks a value in the array as a pivot.
- Divides all elements in the array other than the pivot into two arrays. (those that are larger than the pivot and those that are smaller than the pivot)
- Recursively sorts the two arrays.
- Concats the two arrays and the pivot into one sorted array and returns it.

Examples

quickSort([3,2,1,5,6,7,4]) returns [1, 2, 3, 4, 5, 6, 7]

quickSort([5,9,1,5,6,8,4,2]) returns [1, 2, 4, 5, 5, 6, 8, 9]

quickSort([5,-9,1,5,-6,8,-4,2]) returns [-9, -6, -4, 1, 2, 5, 5, 8]

Edge Cases

If the array has no values in it return an empty array.

The Code

The first thing we need to do inside our function is define three variables. One to store our pivot value, and then two variables that are just empty arrays (one will hold values less than the pivot, the other will hold values greater than the pivot).

The Pivot is an index of your choosing but you do need to make sure its going to be an index that always exists. Since we don't know the length of each array that will be put in I chose the last element in the array to always be our pivot.

```
let quickSort = function(array) {  
  
  let pivot = array[array.length - 1];  
  
  let lows = [];  
  
  let highs = [];  
  
  ...  
  
};
```

Next I have an if statement to handle our edge case

```
// this could have also come before our three variables above if  
// we wanted  
  
if (array.length <= 1) {  
  
  return array;  
  
}
```

Now we need something to iterate through the array and place all elements except our pivot in the lows or highs array. We'll use a trusty for loop

```
// our for loop must stop before reaching the last value in the  
// array because we don't want our pivot placed into either one, so  
// we use array.length-1 in our loop  
  
for (let i = 0; i < array.length-1; i++) {  
  
  // if the value at the index of i is greater than the pivot put  
  // in the highs array  
  
  if (array[i] > pivot) {  
  
    highs.push(array[i]);  
  
  } else {  
  
    // if its lower than or equal to the value of the pivot  
    // then put it in the lows array. When finding an equal  
    // value we could have just as easily decided to put it in  
    // the highs array, the only thing that matters is that  
    // your choice is consistent throughout the algorithm  
  
    lows.push(array[i]);  
  
  }  
  
}
```

Finally its time for some recursive magic

```
// we know from the description above that we need to concat the  
// lows array (after its sorted) with the pivot and the highs array  
// (after its sorted)  
  
// so we call quickSort on the lows array,  
// concat it to the pivot, and concat that with the result of  
// calling quickSort on the highs array  
  
return quickSort(lows).concat(pivot).concat(quickSort(highs))
```

Heres the complete version of the code:

```
let quickSort= function(array) {  
  
  let pivot = array[array.length - 1];  
  
  let lows = [];  
  
  let highs = [];  
  
  if (array.length <= 1) {  
  
    return array;  
  
  }  
  
  for (let i = 0; i < array.length-1; i++) {  
  
    if (array[i] > pivot) {  
  
      highs.push(array[i]);  
  
    } else {  
  
      lows.push(array[i]);  
  
    }  
  
  }  
  
  return quickSort(lows).concat(pivot).concat(quickSort(highs))  
  
};
```

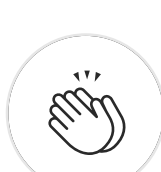
Read More About This Common Sorting Algorithm [HERE](#)

This is just one of many possible solutions, I hope it has helped you gain a better understanding of how to approach this problem. Thank you for reading, and if you found this helpful please leave me a clap!

Find me on GitHub

<https://github.com/JustinPaoletta>

JavaScript Logic Algorithms Problem Solving Code



WRITTEN BY

Justinpaoletta

[Follow](#)

Full Stack Software Engineer. Please reach out if you are

looking to add talent to your team!

ALGORITHM SERIES

Algorithm Series

[Follow](#)



Each week I will post my solution to a computer algorithm I solved and the logic that got me to the solution

[Write the first response](#)

More From Medium

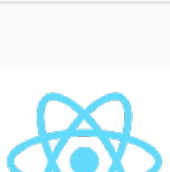
JavaScript Best Practices for Writing More Robust Code — Value Checks

John Au-Yeung in The Startup



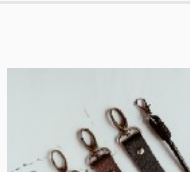
Understanding High-Order Components in React.js (HOCs)

Louis Petrik in Teaching Tech



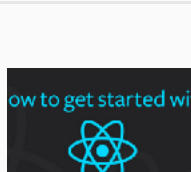
Let's get hooked: a quick introduction to React Hooks

Lekha Surasani



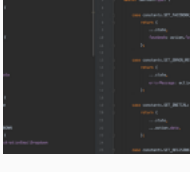
How I learned React JS as a noob — Ultimate React JS Starter Guide

Abdul Wahid Naafi in JavaScript In Plain English



Migrating from CoffeeScript to JavaScript (ES6)

Killian Saint cricq in StuDocu Tech



Making More Complex Requests with the Angular HTTP Client

John Au-Yeung in JavaScript In Plain English



Build a chat app with React, TypeScript and Socket.io

Mihail Gaberov in freeCodeCamp.org



JavaScript Algorithm: Sorted Union

Erica N in The Startup



Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. [Watch](#)

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. [Explore](#)

Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. [Upgrade](#)