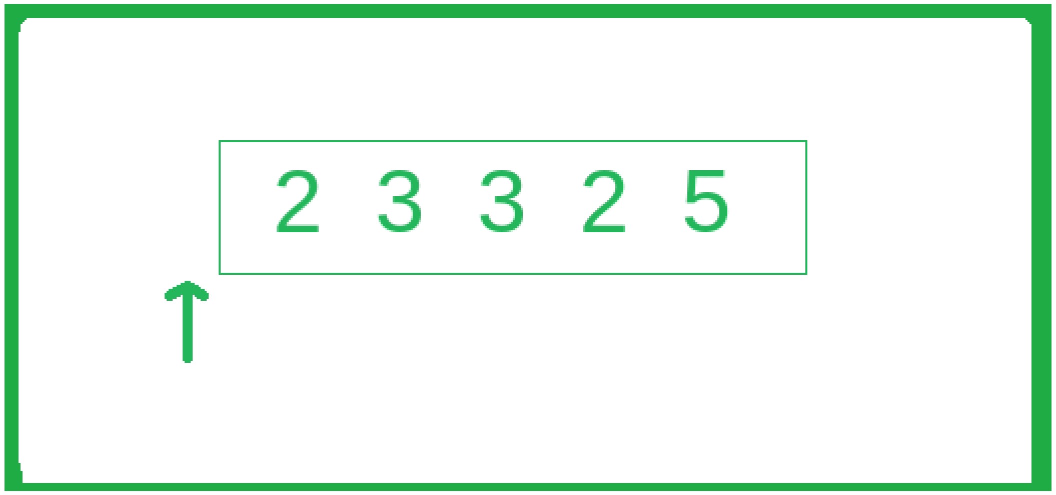
Karan S. Chauhan Follow
Mar 8, 2021 · 4 min read · Listen

Moving Elements To The End of an Array



As I'm studying for technical interviews, I came across a problem that was asking me to move specific elements to the ending of an input array. Since it's a medium level problem , I thought I'd create a blog post about how I approached the problem. Let's start.

Given an array of integers and an integer, create a function that moves all the instances of the given integer to t' 50 | rray. Then, return the modified array.

- **Note: The order of the integers within the array doesn't need to be maintained.

The Approach

The way I think about approaching this is similar to many other data structure and algorithm problem, with pointers.

First, I'd create two variables to represent the positions of the array, one for the beginning and another for the last position of the array.

Then, I can create a conditional statement to invoke only when the two pointers aren't equal. Meaning that the pointer starting from the first position of the array will be before the pointer starting at the ending of the array.

Now to handle an edge case… what if the element at the second pointer (which

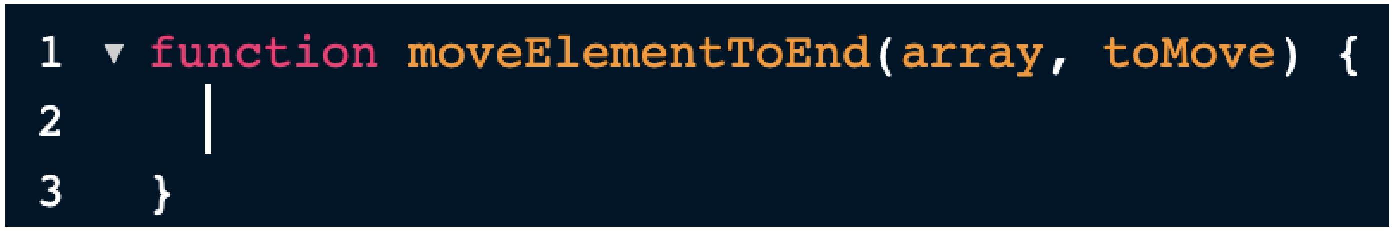
Next, what to do when the element at the firs pointer is equal to the integer I need to move to the end? When this happens, I can swap the elements as needed. This is where I'd create a helper function to handle the swapping functionality.

Then, while we're still in the scope of the initial conditional, increment the first pointer.

Lastly, return the modified array and we should be good. Let's look at the code now.

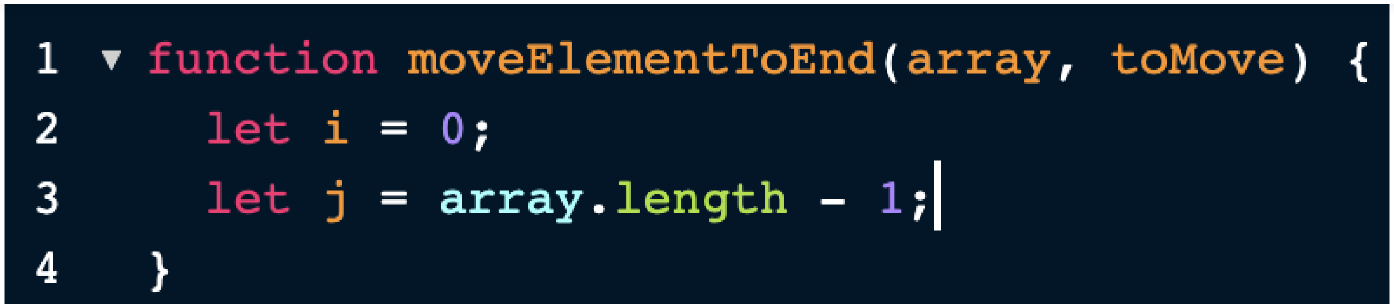
The Code

The function will take two parameters, an array and an integer. Any instance of the integer must be moved to the end.



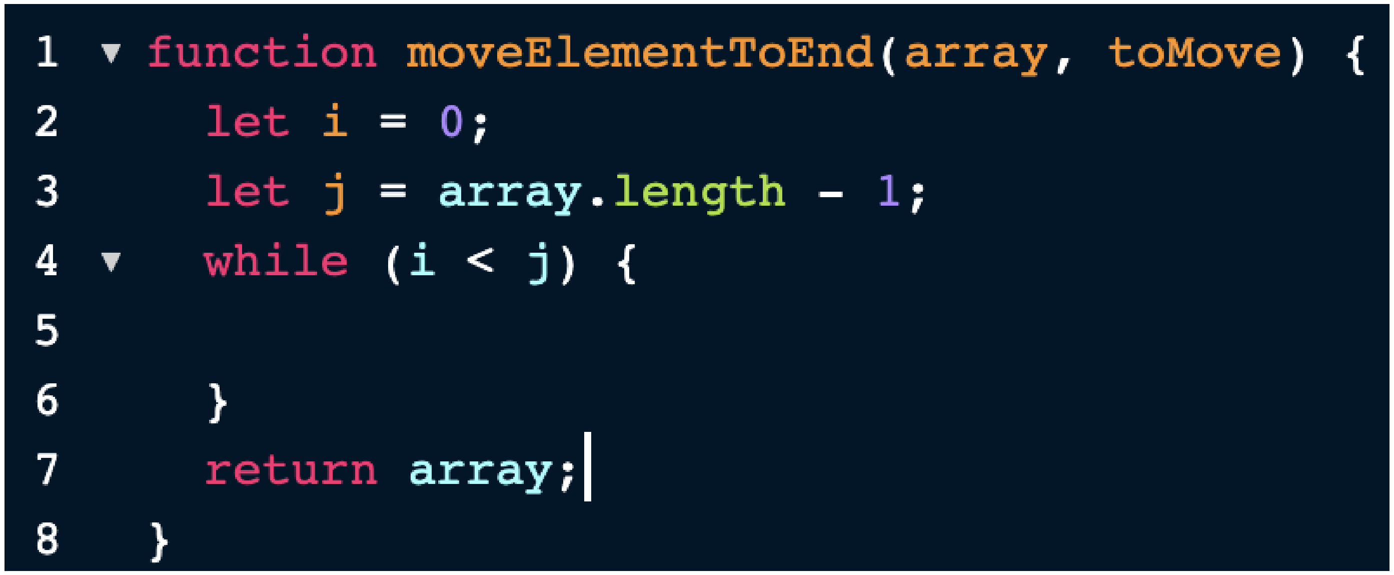
```
function moveElementToEnd(array, toMove) {
    
}
```

Open in appSign In Get started



```
function moveElementToEnd(array, toMove) {
    let i = 0;
    let j = array.length - 1;
}
```

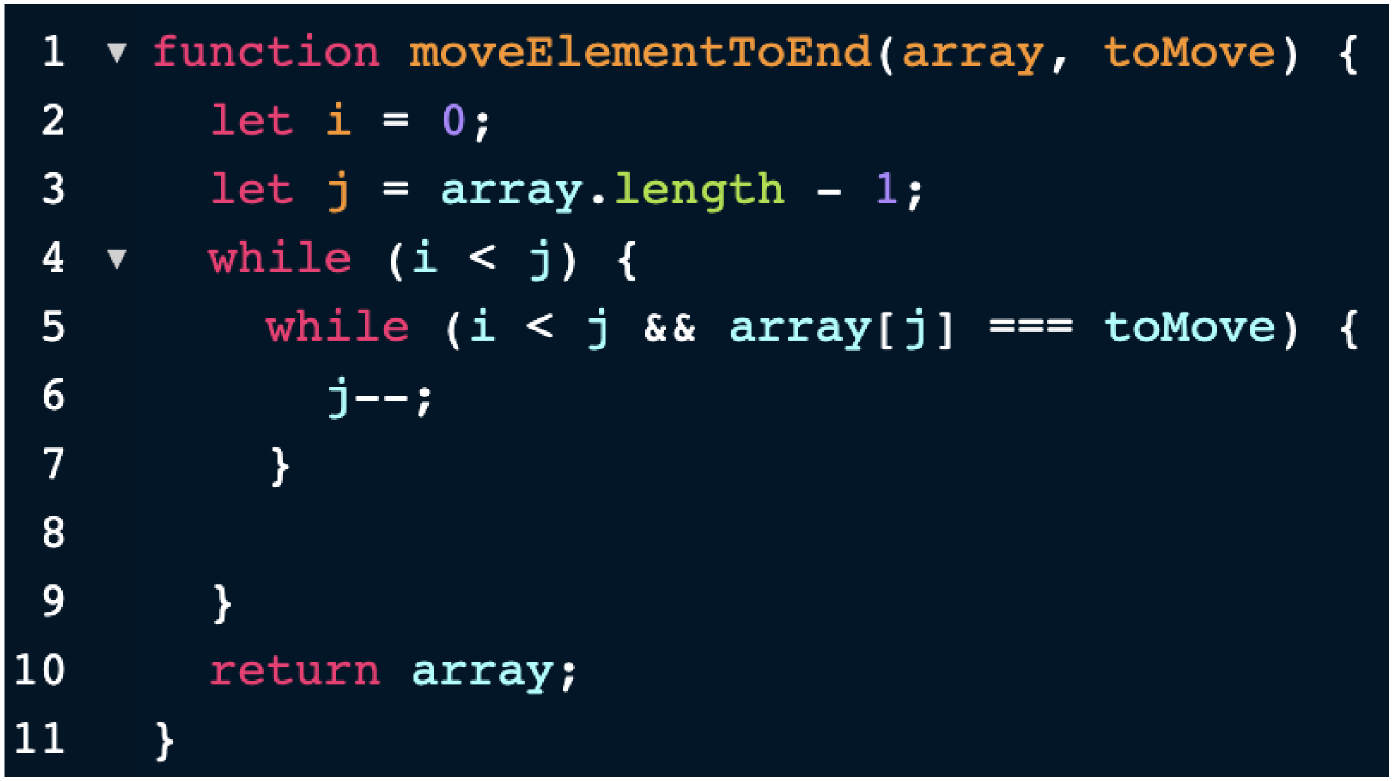
Now we can start on the conditional logic. Let’s use a while loop to invoke whenever “i” is less than “j”. Outside of this while loop, we’ll invoke the final modified array. (The array will be modified within the while loop).



```
function moveElementToEnd(array, toMove) {
    let i = 0;
    let j = array.length - 1;
    while (i < j) {

    }
    return array;
}
```

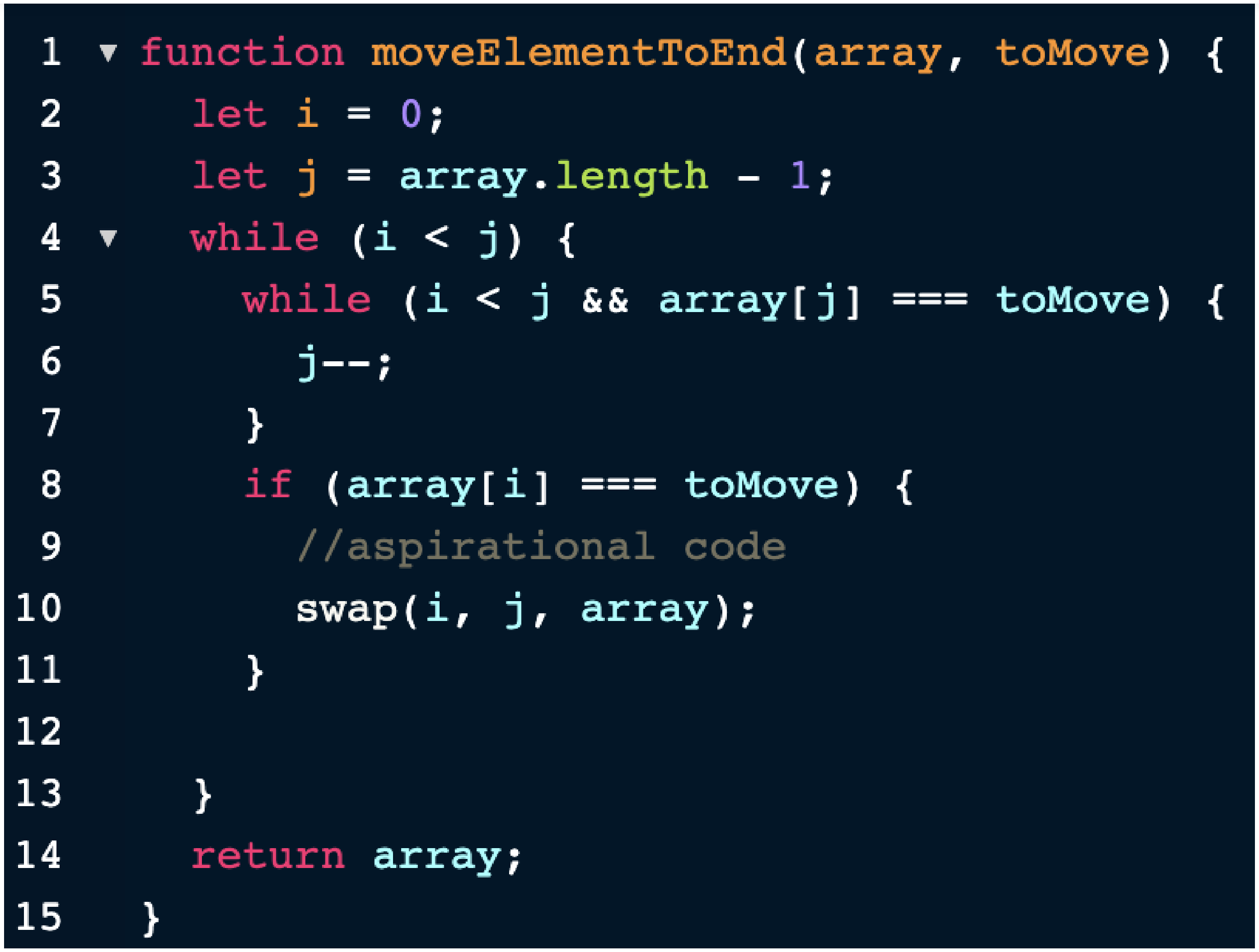
the integer we need to move to the end. If it is, then decrement “j”.



```
function moveElementToEnd(array, toMove) {
    let i = 0;
    let j = array.length - 1;
    while (i < j) {
      while (i < j && array[j] === toMove) {
        j--;
      }

    }
    return array;
}
```

Now after the second while loop, let’s use an if statement to swap elements, this is also where I’ll include aspirational code for a helper method to handle to swapping functionality.

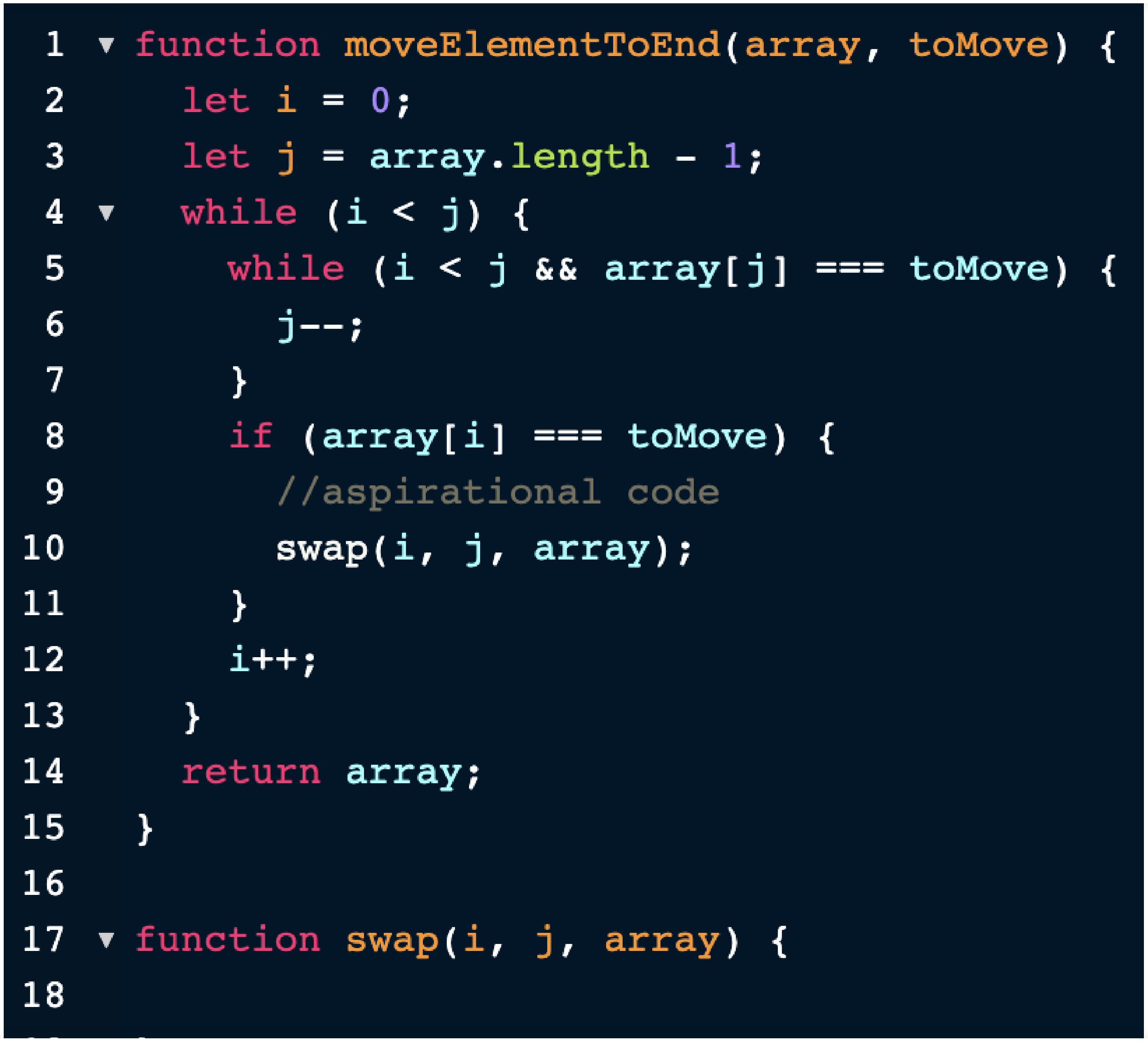


```
function moveElementToEnd(array, toMove) {
  let i = 0;
  let j = array.length - 1;
  while (i < j) {
    while (i < j && array[j] === toMove) {
      j--;
    }
    if (array[i] === toMove) {
      //aspirational code
      swap(i, j, array);
    }

  }
  return array;
}
```

Now within the initial while loop and after the two inner loops (the second while loop and the if statement), let's increment the "i" pointer. Then let's get started

Open in appSign In Get started



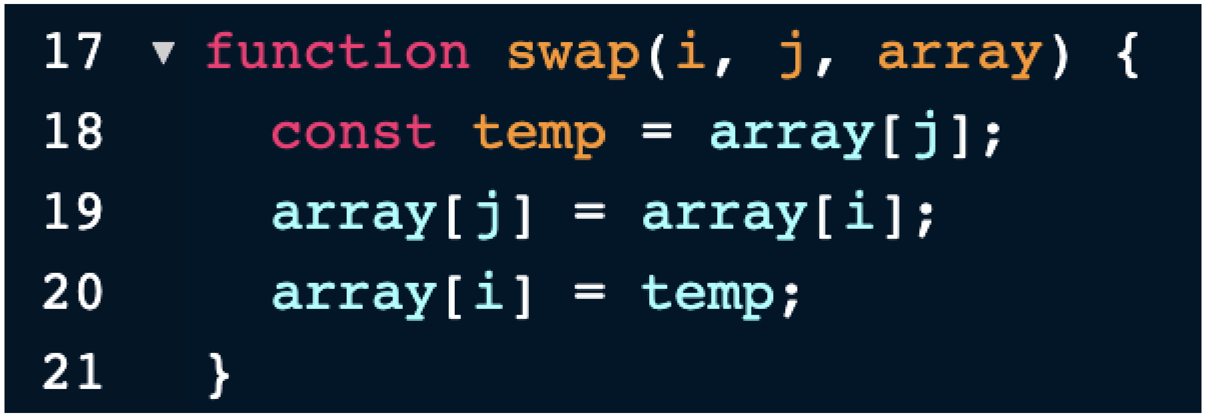
```
function moveElementToEnd(array, toMove) {
  let i = 0;
  let j = array.length - 1;
  while (i < j) {
    while (i < j && array[j] === toMove) {
      j--;
    }
    if (array[i] === toMove) {
      //aspirational code
      swap(i, j, array);
    }
    i++;
  }
  return array;
}

function swap(i, j, array) {

```

- create a temporary variable that will hold the value of the element at index "j"
- on the next line, the element at index "j" will be set to the element at index "i"
- Then, on the next line, the element at index "i" will be set to the temporary variable we initially created.

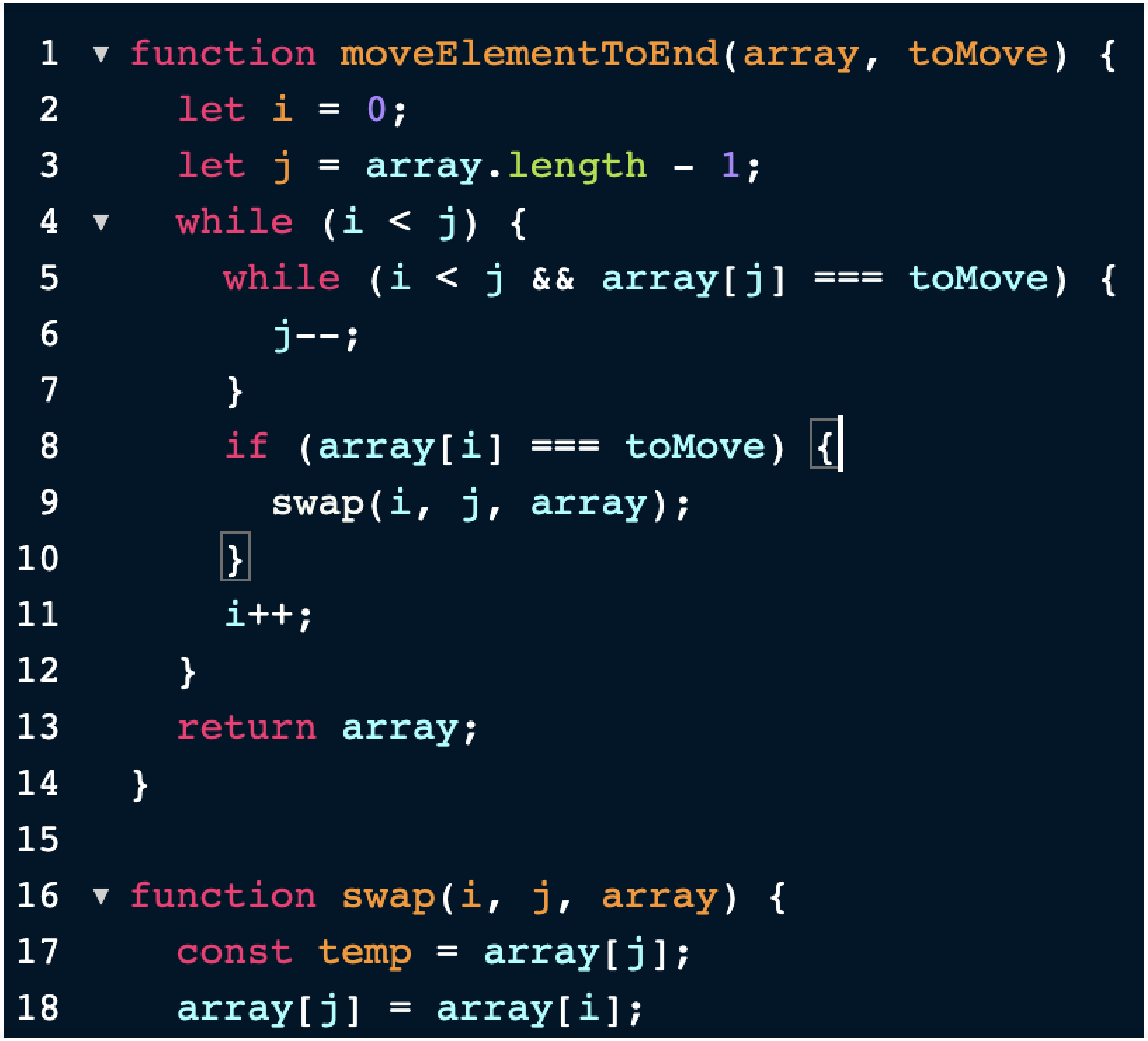
Let's see how this looks.



```
function swap(i, j, array) {
    const temp = array[j];
    array[j] = array[i];
    array[i] = temp;
}
```

And that's it! The final code should look like the following:

Open in app Sign In Get started



```
function moveElementToEnd(array, toMove) {
  let i = 0;
  let j = array.length - 1;
  while (i < j) {
    while (i < j && array[j] === toMove) {
      j--;
    }
    if (array[i] === toMove) {
      swap(i, j, array);
    }
    i++;
  }
  return array;
}

function swap(i, j, array) {
  const temp = array[j];
  array[j] = array[i];
```

Complexity Analysis

The **time complexity** of this solution will be O(n) time, where (n) is the length of the array.

The **space complexity** of this solution is O(1), Constant, space.