Selection Sort

Input: array = [8, 5, 2, 9, 5, 6, 3]

Output: [2, 3, 5, 5, 6, 8, 9]

Input: An array of integers

Output: a sorted version of the input array

Use the selection sort algorithm

```
// O(n^2) time | O(1) space
function selectionSort(array) {
  let currentIndex = 0;
  while (currentIndex < array.length - 1) {
    let smallestIndex = currentIndex;
    for (let i = currentIndex + 1; i < array.length; i++) {
      if (array[smallestIndex] > array[i]) {
        smallestIndex = i;
      }
    }
    [array[currentIndex], array[smallestIndex]] = [
      array[smallestIndex],
      array[currentIndex],
    ];
    currentIndex++;
  }
  return array;
}
```

Time: $O(n^2)$ (where n is the length of the array) since we loop through the array and at every iteration we loop through again trying to find the smallest value. O(n) at best case (if the array is already sorted)

Space: O(1) since we are only swapping values each time

Selection sort works by keeping track of the smallest value in the array and then swapping it with the current value (which begins at index 0) until the array is sorted

We go to array.length - 1 since, when we get to the end, the last value will be in it's sorted position already