



Code Challenge #13 Selection Sort (Easy)

Difficulty:  Category:  Successful Submissions: 35,718+

Selection Sort

Write a function that takes in an array of integers and returns a sorted version of that array. Use the Selection Sort algorithm to sort the array.

If you're unfamiliar with Selection Sort, we recommend watching the Conceptual Overview section of this question's video explanation before starting to code.

Sample Input

```
array = [8, 5, 2, 9, 5, 6, 3]
```

Sample Output

```
[2, 3, 5, 5, 6, 8, 9]
```

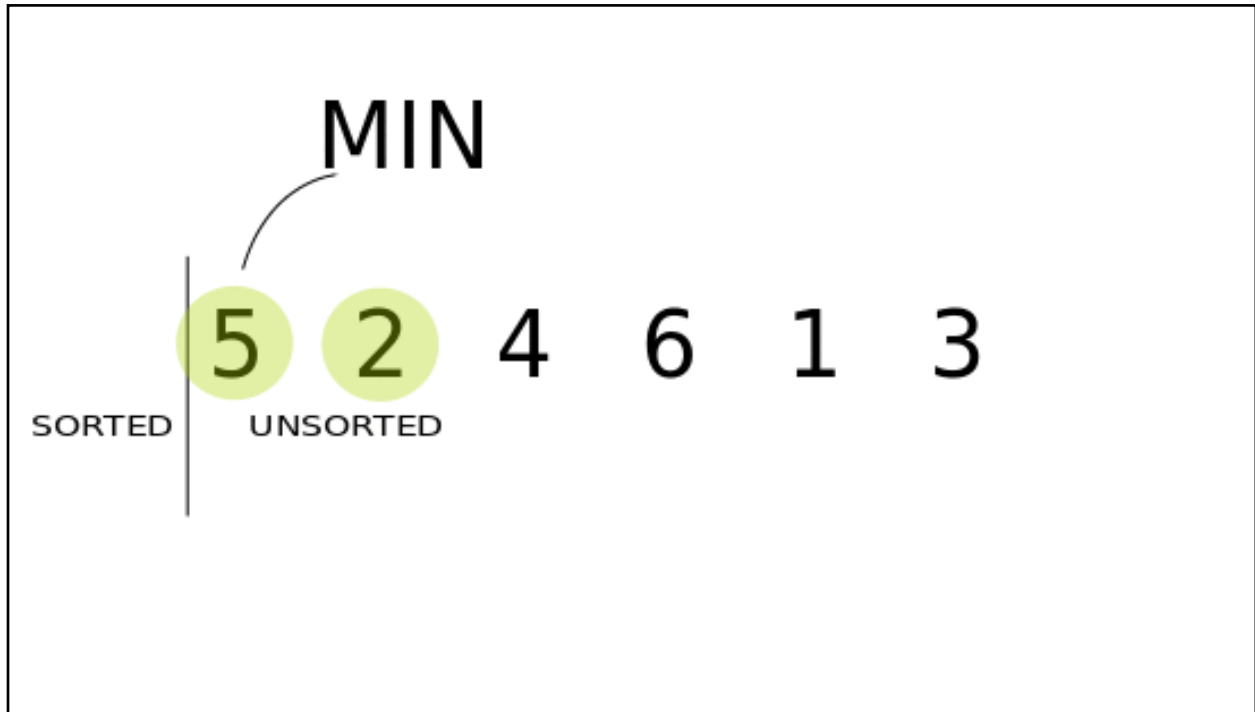
Solution #1

```
1. function selectionSort(array) {
2.   let startIdx = 0;
3.   while (startIdx < array.length - 1) {
4.     let smallestIdx = startIdx;
5.     for (let i = startIdx + 1; i < array.length; i++) {
6.       if (array[smallestIdx] > array[i]) smallestIdx = i;
7.     }
8.     swap(startIdx, smallestIdx, array);
9.     startIdx++;
10.  }
11.  return array
12. }
13.
14. function swap(i, j, array) {
15.   const temp = array[j];
16.   array[j] = array[i];
17.   array[i] = temp;
18. }
```

Explanation

Selection sort is a sorting algorithm where we find the smallest element in an array first. Once we find the smallest element, we exchange that element with the first

element in the array. We then find the second smallest element and exchange that element with the element at the second position. We continue this process till we have an sorted array. See animated gif below.



The solution for this problem begins with creating a let variable called `startIdx` which has an initial value of 0. We then use a while loop which loops through as long as `startIdx < array.length - 1`. Within the while loop we have a variable called `smallestIdx = startIdx`. We do this because we assume the smallest element in the array is the first item in the array (even though it may not be). We then have a for loop. Inside the for loop we have a let variable called `i` which is equal to the `startIdx + 1`, as long as `i < array.length` we increment `i` by 1. Within the for loop we check to see if the left item `array[smallestIdx] > array[i]`. If it is we assign `i` to `smallestIdx`. We do this because we want to find the smallest item in the array by going through all the items in the array. Once the for loop finishes we use a helper method called `swap` to switch the first item in array with the smallest item in the array. (The swap helper method arguments are different from the implementation of the helper method. The swap method inside the `selectionSort` functions has three arguments labeled as `startIdx`, `smallestIdx` and `array`.) We then increment `startIdx` by 1 to continue testing the next item in array. Once all the checking is complete we return the array. The helper function called `swap` is a simple function that takes in three arguments. The three arguments are `i`, `j` and

array. Inside the helper function we assign the value of `array[j]` to a const variable called `temp`. We then assign the value of `array[i]` to `array[j]`. We then assign the value of variable `temp` to `array[i]`. The Big O is $O(n^2)$ for worst case scenario and average case. The best case is $O(n)$ is also $O(n^2)$.