

## Lesson 04 Demo 02

### Implementing Selection Sort Algorithm

**Objective:** To sort an array using selection sort in JavaScript, understanding its time and space complexity for organizing fixed-size datasets such as weekly task lists or student scores

**Tools required:** Visual Studio Code and Node.js

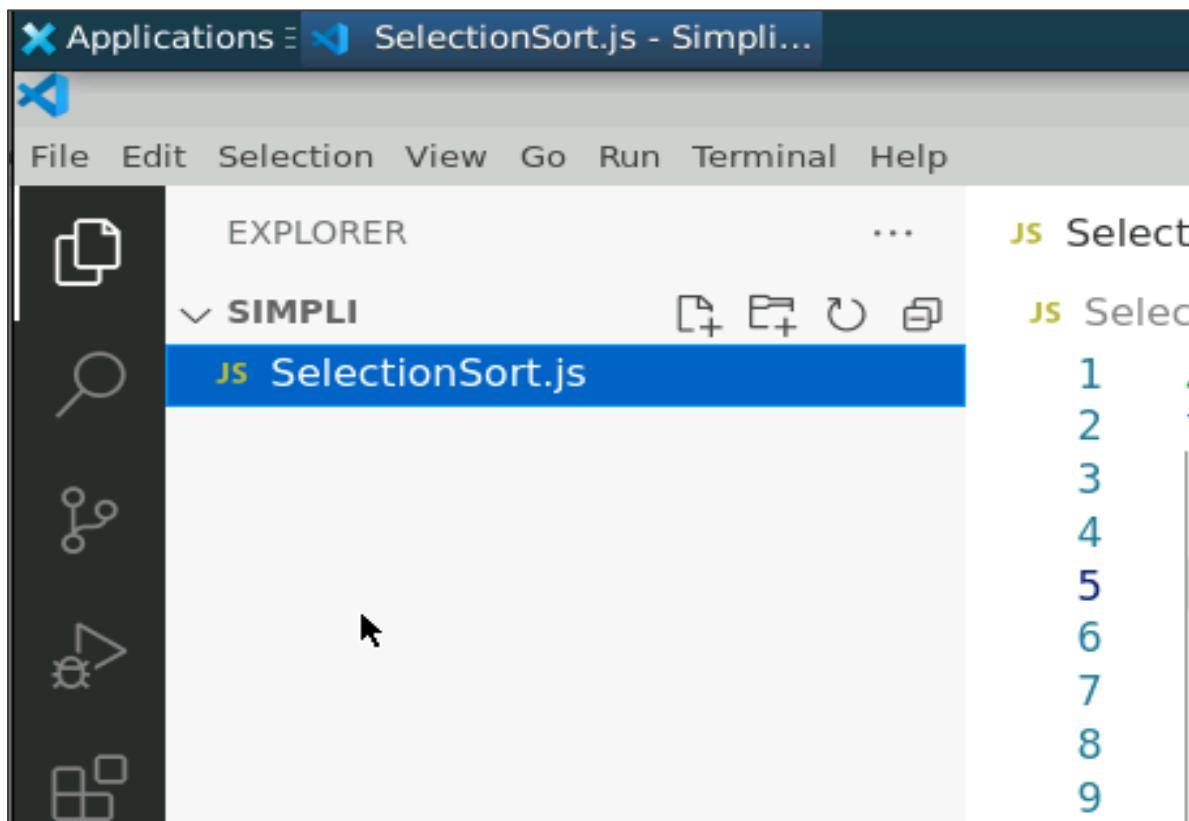
**Prerequisites:** A basic understanding of arrays and loops in JavaScript

Steps to be followed:

1. Create a JavaScript file and execute it

#### Step 1: Create a JavaScript file and execute it

1.1 Open the Visual Studio Code editor and create a JavaScript file named **SelectionSort.js**



1.2 Add the following code to the file:

```
// Function to perform selection sort on an array
function selectionSort(array) {
    // Time Complexity: O(n^2) - Quadratic time complexity
    // Space Complexity: O(1) - Constant space complexity

    // Iterate through each element in the array
    for (let i = 0; i < array.length; i++) {
        // Assume the current index is the minimum value index
        let minValueIndex = i;

        // Find the index of the minimum value in the remaining unsorted part of the array
        for (let j = i + 1; j < array.length; j++) {
            if (array[j] < array[minValueIndex]) {
                minValueIndex = j;
            }
        }

        // Swap the current element with the minimum value element
        [array[i], array[minValueIndex]] = [array[minValueIndex], array[i]];
    }

    return array;
}

// Example usage
const unsortedArray = [5, 2, 4, 1, 3];

// Measure the execution time of the selectionSort function
console.time("selectionSort");
const sortedArray = selectionSort(unsortedArray);
console.timeEnd("selectionSort");

console.log(sortedArray);
```

```
1 // Function to perform selection sort on an array
2 function selectionSort(array) {
3     // Time Complexity: O(n^2) - Quadratic time complexity
4     // Space Complexity: O(1) - Constant space complexity
5
6     // Iterate through each element in the array
7     for (let i = 0; i < array.length; i++) {
8         // Assume the current index is the minimum value index
9         let minValueIndex = i;
10
11        // Find the index of the minimum value in the remaining unsorted part of the array
12        for (let j = i + 1; j < array.length; j++) {
13            if (array[j] < array[minValueIndex]) {
14                minValueIndex = j;
15            }
16        }
17
18        // Swap the current element with the minimum value element
19        [array[i], array[minValueIndex]] = [array[minValueIndex], array[i]];
20    }
21
22    return array;
23}
24
25 // Example usage
26 const unsortedArray = [5, 2, 4, 1, 3];
27
28 // Measure the execution time of the selectionSort function
29 console.time("selectionSort");
30 const sortedArray = selectionSort(unsortedArray);
31 console.timeEnd("selectionSort");
32
33
```

```
25 // Example usage
26 const unsortedArray = [5, 2, 4, 1, 3];
27
28 // Measure the execution time of the selectionSort function
29 console.time("selectionSort");
30 const sortedArray = selectionSort(unsortedArray);
31 console.timeEnd("selectionSort");
32
33 console.log(sortedArray);
```

1.3 Press **Ctrl + S** to save the file and then execute it in the **TERMINAL** using the following commands:

```
ls  
node SelectionSort.js
```

The screenshot shows a code editor interface with a terminal tab selected. The code editor displays a portion of a JavaScript file named `SelectionSort.js`. The terminal window shows the command `node SelectionSort.js` being run, followed by the output of the sorted array [1, 2, 3, 4, 5] and the execution time of 0.104ms.

```
5 // Iterate through each element in the array
6 for (let i = 0; i < array.length; i++) {
7     // Assume the current index is the minimum value index
8     let minValueIndex = i;
9
10    // Find the index of the minimum value in the remaining unsorted part
11    for (let j = i + 1; j < array.length; j++) {
12        if (array[j] < array[minValueIndex]) {
13            minValueIndex = j;
14        }
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL >
priyanshurajsim@ip-172-31-40-74:~/Downloads/Simpli$ ls
SelectionSort.js
priyanshurajsim@ip-172-31-40-74:~/Downloads/Simpli$ node SelectionSort.js
selectionSort: 0.104ms
[ 1, 2, 3, 4, 5 ]
priyanshurajsim@ip-172-31-40-74:~/Downloads/Simpli$
```

By following these steps, you have successfully used the selection sort algorithm in JavaScript to organize small datasets like task lists or scores, understood how it arranges data through comparisons, and learned that it has a time complexity of  $O(n^2)$  and space complexity of  $O(1)$ .