Lesson 02 Demo 02

Maximum Subarray Sum

Objective: To find the highest sum from any length subarray, including zero, in an array of integers with both negative and positive values

Tools required: Visual Studio Code (VS Code) and JavaScript

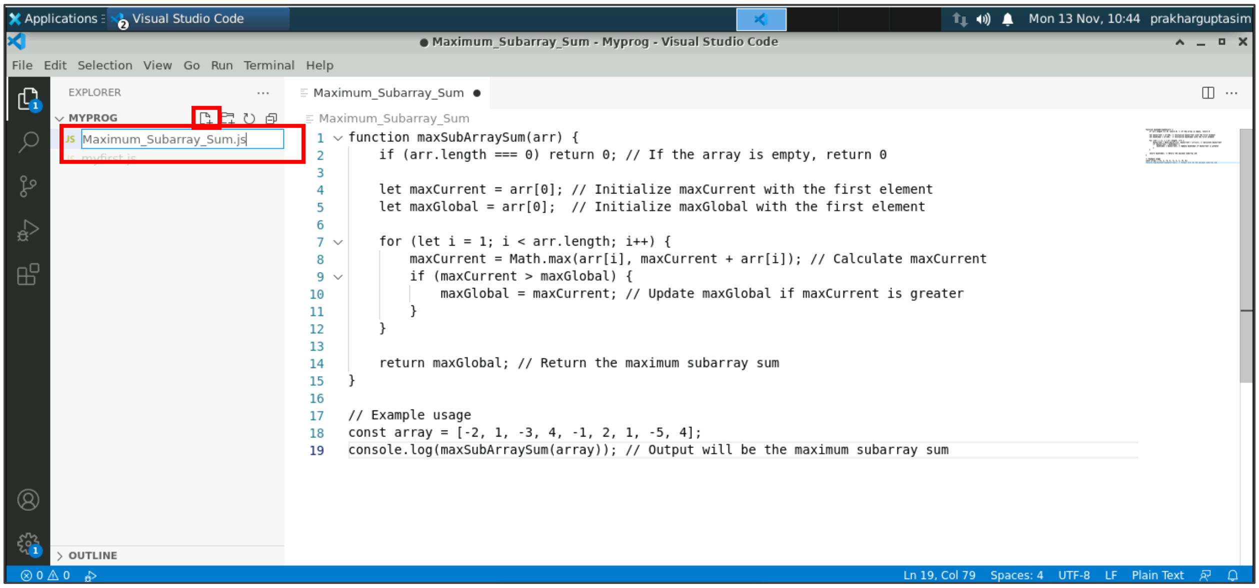
Prerequisites: Perform demo 01 of lesson 02

Steps to be followed:

1. Create and execute the JS file

Step 1: Create and execute the JS file

1.1 Create a JavaScript file named **Maximum_Subarray_Sum.js** by clicking on the **New File** icon as shown below:

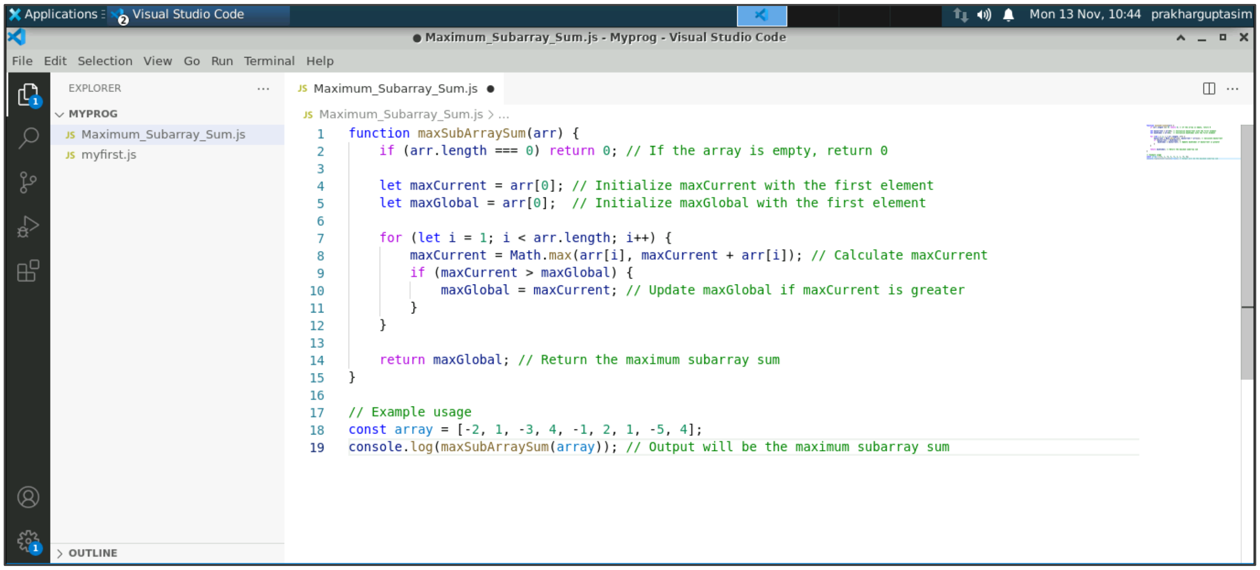


Applications
Visual Studio Code
Mon 13 Nov, 10:44 prakharguptasim
● Maximum_Subarray_Sum - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
Maximum_Subarray_Sum.js
Maximum_Subarray_Sum ●
Maximum_Subarray_Sum
OUTLINE
Ln 19, Col 79 Spaces: 4 UTF-8 LF Plain Text

```
function maxSubArraySum(arr) {
    if (arr.length === 0) return 0; // If the array is empty, return 0

    let maxCurrent = arr[0]; // Initialize maxCurrent with the first element
    let maxGlobal = arr[0];  // Initialize maxGlobal with the first element

    for (let i = 1; i < arr.length; i++) {
        maxCurrent = Math.max(arr[i], maxCurrent + arr[i]); // Calculate maxCurrent
        if (maxCurrent > maxGlobal) {
            maxGlobal = maxCurrent; // Update maxGlobal if maxCurrent is greater
        }
    }

    return maxGlobal; // Return the maximum subarray sum
}

// Example usage
const array = [-2, 1, -3, 4, -1, 2, 1, -5, 4];
console.log(maxSubArraySum(array)); // Output will be the maximum subarray sum
```

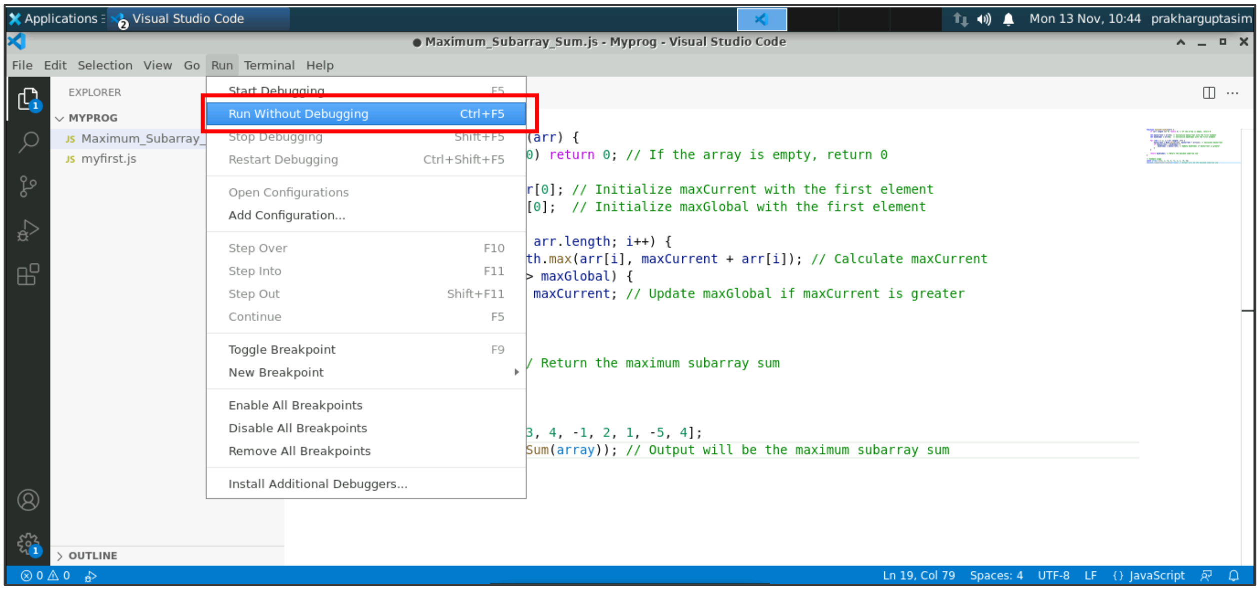
1.2 Write the code given below in the file created in step 1.1:

```
function maxSubArraySum(arr) {
if (arr.length === 0) return 0; // If the array is empty, return 0
let maxCurrent = arr[0]; // Initialize maxCurrent with the first element
let maxGlobal = arr[0];  // Initialize maxGlobal with the first element

for (let i = 1; i < arr.length; i++) {
maxCurrent = Math.max(arr[i], maxCurrent + arr[i]); // Calculate maxCurrent
if (maxCurrent > maxGlobal) {
maxGlobal = maxCurrent; // Update maxGlobal if maxCurrent is greater
}
}
return maxGlobal; // Return the maximum subarray sum
}
// Example usage
const array = [-2, 1, -3, 4, -1, 2, 1, -5, 4];
console.log(maxSubArraySum(array)); // Output will be the maximum subarray sum
```

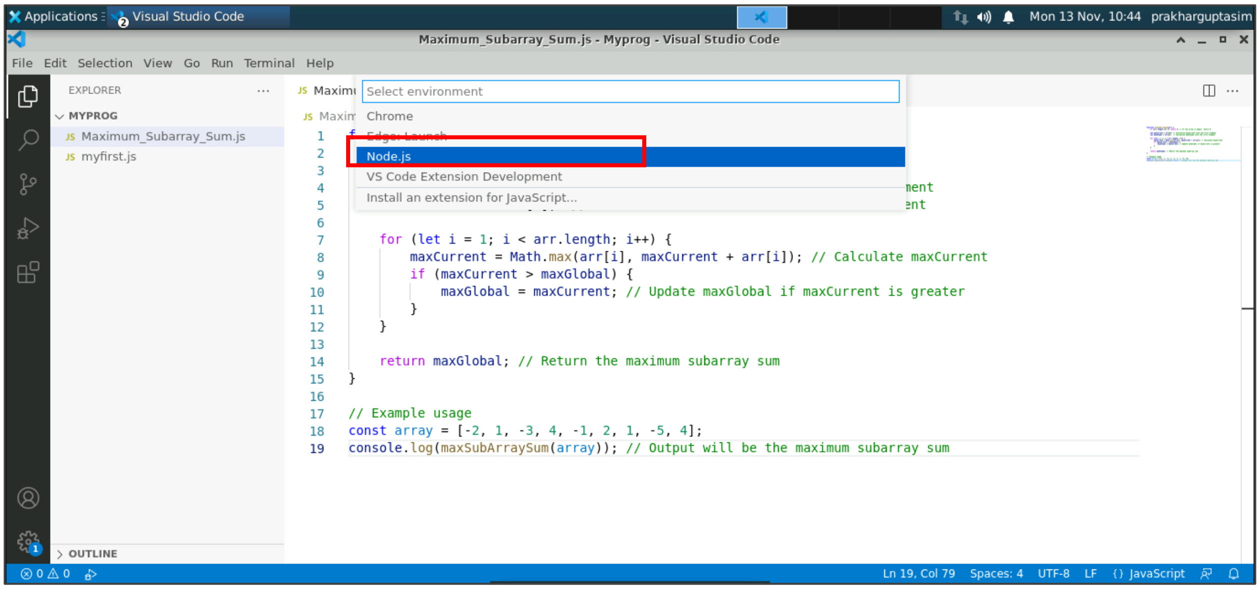

Applications
Visual Studio Code
Mon 13 Nov, 10:44 prakharguptasim
● Maximum_Subarray_Sum.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
Maximum_Subarray_Sum.js
myfirst.js
OUTLINE
Maximum_Subarray_Sum.js ●
Maximum_Subarray_Sum.js > ...

```
function maxSubArraySum(arr) {
    if (arr.length === 0) return 0; // If the array is empty, return 0

    let maxCurrent = arr[0]; // Initialize maxCurrent with the first element
    let maxGlobal = arr[0];  // Initialize maxGlobal with the first element

    for (let i = 1; i < arr.length; i++) {
        maxCurrent = Math.max(arr[i], maxCurrent + arr[i]); // Calculate maxCurrent
        if (maxCurrent > maxGlobal) {
            maxGlobal = maxCurrent; // Update maxGlobal if maxCurrent is greater
        }
    }

    return maxGlobal; // Return the maximum subarray sum
}

// Example usage
const array = [-2, 1, -3, 4, -1, 2, 1, -5, 4];
console.log(maxSubArraySum(array)); // Output will be the maximum subarray sum
```

1.3 Save the code and click on **Run->Run Without Debugging->Node.js** to check the output in the debug console

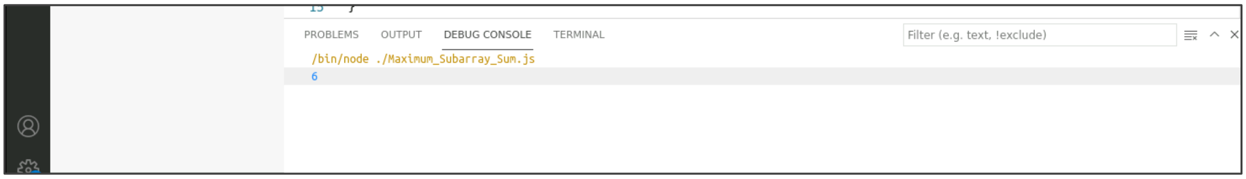


```
(arr) {
0) return 0; // If the array is empty, return 0

r[0]; // Initialize maxCurrent with the first element
[0];  // Initialize maxGlobal with the first element

 arr.length; i++) {
th.max(arr[i], maxCurrent + arr[i]); // Calculate maxCurrent
> maxGlobal) {
 maxCurrent; // Update maxGlobal if maxCurrent is greater


/ Return the maximum subarray sum



3, 4, -1, 2, 1, -5, 4];
Sum(array)); // Output will be the maximum subarray sum
```


Applications
Visual Studio Code
Mon 13 Nov, 10:44 prakharguptasim
Maximum_Subarray_Sum.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
MYPROG
Maximum_Subarray_Sum.js
myfirst.js
Maximu
Select environment
Chrome
Edge: Launch
Node.js
VS Code Extension Development
Install an extension for JavaScript...
OUTLINE
Ln 19, Col 79 Spaces: 4 UTF-8 LF {} JavaScript

```
    for (let i = 1; i < arr.length; i++) {
        maxCurrent = Math.max(arr[i], maxCurrent + arr[i]); // Calculate maxCurrent
        if (maxCurrent > maxGlobal) {
            maxGlobal = maxCurrent; // Update maxGlobal if maxCurrent is greater
        }
    }

    return maxGlobal; // Return the maximum subarray sum
}

// Example usage
const array = [-2, 1, -3, 4, -1, 2, 1, -5, 4];
console.log(maxSubArraySum(array)); // Output will be the maximum subarray sum
```

- Now you see the output in the debug console as shown below:



```
/bin/node ./Maximum_Subarray_Sum.js
6
```

Explanation:

1. maxCurrent: Tracks the highest sum of the subarray ending at the current index
2. maxGlobal: Tracks the highest sum found overall
3. The loop begins from the second element (index 1) as the first element is already accounted for in both maxCurrent and maxGlobal.
4. At each element, a decision is made to extend the current subarray (maxCurrent + arr[i]) or begin a new subarray from this element (arr[i]), choosing the larger option using Math.max(arr[i], maxCurrent + arr[i]).
5. If maxCurrent surpasses maxGlobal, maxGlobal is updated.
6. Following the loop, maxGlobal contains the highest sum of any contiguous subarray.

Conclusion:

By following the above steps, you have successfully found the subarray having the maximum sum.