

Lesson 02 Demo 03 Balanced Array

Objective: To ascertain whether an array of even length can be divided into two halves with equal sums and unique elements in each half

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

Prerequisites: Perform demo 01 of lesson 02

Steps to be followed:

1. Create the algorithm and run it

Step 1: Create the algorithm and run it

1.1 Create a JavaScript file named Balanced_Array.js shown below:

```
🕻 Applications 🗄 🌏 Visual Studio Code
                                ··· Js Balanced_Array.js ●
         MYPROG
                                                  JS Balanced_Array.js > 😭 isBalancedArray
                                                          function isBalancedArray(arr) {
   if (arr.length % 2 !== 0) return false; // Array length must be even
          Maximum Subarray Sum.js
         Js myfirst.js
                                                                let totalSum = arr.reduce((acc, cur) => acc + cur, 0);
                                                                if (totalSum % 2 !== 0) return false; // Total sum must be even for a balanced array
                                                                let halfSum = totalSum / 2;
                                                    10
11
                                                                let currentSum = 0;
                                                                for (let i = 0; i < arr.length; i++) {
    currentSum += arr[i];
    set.add(arr[i]);</pre>
                                                    14
15
16
                                                                       if (currentSum === halfSum) {
                                                                            // Check if the remaining elements are distinct
let remainingElements = arr.slice(i + 1);
let remainingSet = new Set(remainingElements);
                                                                            if (remainingSet.size === remainingElements.length) {
                                                                            return true; // Both halves are balanced and have unique elements
} else {
return false; // Second half has duplicates
```



1.2 Paste the code in the file created in step 1.1 as shown below: function isBalancedArray(arr) { if (arr.length % 2 !== 0) return false; // Array length must be even let totalSum = arr.reduce((acc, cur) => acc + cur, 0); if (totalSum % 2 !== 0) return false; // Total sum must be even for a balanced array let halfSum = totalSum / 2; let set = new Set(); let currentSum = 0; for (let i = 0; i < arr.length; i++) { currentSum += arr[i]; set.add(arr[i]); if (currentSum === halfSum) { // Check if the remaining elements are distinct let remainingElements = arr.slice(i + 1); let remainingSet = new Set(remainingElements); if (remainingSet.size === remainingElements.length) { return true; // Both halves are balanced and have unique elements } else { return false; // Second half has duplicates } return false; // No balanced division found

```
S Balanced_Array.js 

•
JS Balanced Array.js > 😭 isBalancedArray
 function isBalancedArray(arr) {
         if (arr.length % 2 !== 0) return false; // Array length must be even
         let totalSum = arr.reduce((acc, cur) => acc + cur, 0);
         if (totalSum % 2 !== 0) return false; // Total sum must be even for a balanced array
         let halfSum = totalSum / 2:
         let set = new Set():
         let currentSum = 0;
10
11
         for (let i = 0; i < arr.length; i++) {
12
             currentSum += arr[i];
13
             set.add(arr[i]);
14
15
             if (currentSum === halfSum) {
16
                 // Check if the remaining elements are distinct
17
                  let remainingElements = arr.slice(i + 1);
18
                 let remainingSet = new Set(remainingElements);
19
20
                  if (remainingSet.size === remainingElements.length) {
21
                     return true; // Both halves are balanced and have unique elements
22
23
24
                      return false; // Second half has duplicates
```

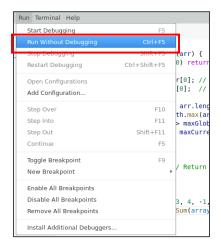
console.log(isBalancedArray(array)); // Output will depend on the input array

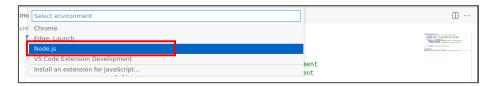
// Example usage

const array = [1, 2, 3, 4, 5, 6];



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





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

```
Balanced_Array.js - Myprog - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Ð
                                  ... Js Balanced_Array.js ×
                                                                                                                                                                                V MYPROG
                                          JS Balanced_Array.js > ♦ isBalancedArray
                                                function isBalancedArray(arr) {
   if (arr.length % 2 !== 0) return false; // Array length must be even
      Js Balanced_Array.js
       Js Maximum_Subarray_Sum.js
       Js myfirst.js
                                                     let totalSum = arr.reduce((acc, cur) => acc + cur, 0);
                                                     if (totalSum % 2 !== 0) return false; // Total sum must be even for a balanced array
                                                      let halfSum = totalSum / 2;
                                                      let set = new Set();
                                           10
11
                                                     let currentSum = 0;
                                                      for (let i = 0; i < arr.length; i++) {
    currentSum += arr[i];</pre>
                                           12
13
                                                          set.add(arr[i]):
                                                                                                                                    Filter (e.g. text, !exclude)
                                                                                                                                                                              ≡ ^ ×
                                            /bin/node ./Balanced_Array.js
```



Explanation:

- 1. First, we check if the total sum of the array is even and the length of the array is even.
- 2. We calculate the half sum, which is what each half should sum up to for the array to be balanced.
- 3. As we iterate through the array, we keep adding elements to a set and summing up their values.
- 4. If, at any point, the current sum equals the half sum, we check the remaining elements in the array. If these remaining elements are all unique (checked by comparing the size of a set constructed from these elements with the length of these elements), then the array can split into two balanced halves. Otherwise, it cannot.

Conclusion:

By following the above step, you have successfully checked whether the given array is balanced or not.