

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(n) time | O(d) space - where n is the total number of elements in
4 // including sub-elements, and d is the greatest depth of "special" ar
5 function productSum(array, multiplier = 1) {
6   let sum = 0;
7   for (const element of array) {
8     if (Array.isArray(element)) {
9       sum += productSum(element, multiplier + 1);
10    } else {
11      sum += element;
12    }
13  }
14  return sum * multiplier;
15 }
16
17 exports.productSum = productSum;
18
```

Solution 1

Solution 2

Solution 3

```
1 function productSum(array) {
2   // Write your code here.
3 }
4
5 // Do not edit the line below.
6 exports.productSum = productSum;
7
```

Custom Output

Submit Code

Run or submit code when you're ready.

Our Tests

```
1 assert isinstance(my_list, list)
2 assert len(my_list) == 5
3
4 assert isinstance(my_list[0], int)
5 assert isinstance(my_list[1], int)
6 assert isinstance(my_list[2], int)
7 assert isinstance(my_list[3], int)
8 assert isinstance(my_list[4], int)
```