

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(n) time | O(1) space - where n is the length of the input array
4 function hasSingleCycle(array) {
5   let numElementsVisited = 0;
6   let currentIdx = 0;
7   while (numElementsVisited < array.length) {
8     if (numElementsVisited > 0 && currentIdx === 0) return false;
9     numElementsVisited++;
10    currentIdx = getNextIdx(currentIdx, array);
11  }
12  return currentIdx === 0;
13 }
14
15 function getNextIdx(currentIdx, array) {
16   const jump = array[currentIdx];
17   const nextIdx = (currentIdx + jump) % array.length;
18   return nextIdx >= 0 ? nextIdx : nextIdx + array.length;
19 }
20
21 exports.hasSingleCycle = hasSingleCycle;
22
```

Solution 1 Solution 2 Solution 3

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

Run or submit code when you're ready.