

Our Solution(s)

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n) time | O(1) space
5     func hasSingleCycle(array: [Int]) -> Bool {
6         var currentIndex = 0
7         var numberOfElementsVisited = 0
8
9         var innerArray = array
10        while numberOfElementsVisited < array.count {
11            if currentIndex == 0, numberOfElementsVisited > 0 {
12                return false
13            }
14
15            numberOfElementsVisited += 1
16            currentIndex = getNextIndex(&currentIndex, array: &
17        }
18
19        return currentIndex == 0
20    }
21
22    func getNextIndex(_ currentIndex: inout Int, array: inout [I
23        let jump = array[currentIndex]
24        let nextIndex = (jump + currentIndex) % array.count
25
26        if nextIndex >= 0 {
27            return nextIndex
28        } else {
29            return nextIndex + array.count
30        }
31    }
32 }
33
```

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     func hasSingleCycle(array: [Int]) -> Bool {
3         // Write your code here.
4         return false
5     }
6 }
7
```

Custom Output

Submit Code

